



UNIVERSITI MALAYA

Perpustakaan SKTM

SISTEM INFORMASI DIREKTORI BAS

Disediakan oleh:

NIK EZAMI BIN NIK ISMAIL

WEK 990395

SESI 2002/2003

**PROJEK ILMIAH
TAHAP AKHIR II
WXES3182**

Penyelia:

DR. RUKAINI HJ ABDULLAH

Moderator:

ENCIK NOR RIDZUAN DAUD

ABSTRAK

Sistem Informasi Direktori Bas merupakan satu sistem informasi berkomputer yang memudahkan proses memperolehi maklumat mengenai bas. Sistem ini akan digunakan oleh pengguna-pengguna bas yang ingin merancang perjalanan mereka atau pengguna yang hanya mahukan maklumat secara langsung mengenai bas. Mereka boleh mengakses sistem ini dimana sahaja dan tidak kira waktu mengikut kehendak mereka dengan syarat ada sambungan ke internet.

Sistem ini diharap dapat merancang perjalanan pengguna-pengguna bas dikawasan Kuala Lumpur/Selangor dengan tepat dan sistematik. Perancangan yang dibuat oleh sistem adalah berdasarkan kepada laluan terpantas antara 2 stesyen/perhentian bas. Sistem manual yang sedia ada adalah agak terhad dari segi kewujudan. Sebagai contoh, papan tanda yang mempunyai peta rujukan tidak wujud di setiap perhentian bas. Sistem juga membenarkan pengemaskinian dijalankan dengan mudah tanpa kos yang tinggi.

ISI KANDUNGAN

| | |
|----------------|-----|
| Abstrak | i |
| Penghargaan | ii |
| Kandungan | iii |
| Senarai Rajah | ix |
| Senarai Jadual | xi |

BAB 1: PENGENALAN

| | |
|----------------------------|---|
| 1.1 Pengenalan Projek | 1 |
| 1.2 Masalah yang wujud | 3 |
| 1.3 Masalah yang dihadapi | 4 |
| 1.4 Objektif | 4 |
| 1.5 Skop Projek | 5 |
| 1.6 Sasaran Pengguna | 6 |
| 1.7 Perancangan Projek | 7 |
| 1.8 Hasil Yang Dijangkakan | 9 |

BAB 2: KAJIAN LITERASI

Bahagian I: Kajian dan Penulisan

| | |
|---------------------|----|
| 2.1 Tujuan | 11 |
| 2.2 Definisi Sistem | 12 |

| | | |
|-------|--|----|
| 2.3 | Penemuan Rujukan/Kaedah Kajian | 13 |
| 2.3.1 | Kaedah Pengumpulan Data | 13 |
| 2.3.2 | Kaedah Penulisan | 15 |
| 2.4 | Kajian Berkaitan Algoritma Shortest Path | 16 |
| 2.4.1 | Heuristic Search | 16 |
| 2.4.2 | Algoritma Dijkstra | 24 |
| 2.5 | Kelemahan Sistem Manual | 29 |
| 2.6 | Kelebihan Sistem Informasi Atas Talian | 30 |

Bahagian II: Kajian Sistem Sedia Ada

| | | |
|-----|---|----|
| 2.7 | Bus Information System | 33 |
| 2.8 | Madrid Region Government Transport Information System | 38 |

BAB 3: METODOLOGI PEMBANGUNAN SISTEM

| | | |
|---------|-------------------------------|----|
| 3.1 | Pengenalan | 42 |
| 3.2 | Metodologi | 43 |
| 3.2.1 | Fasa I: Kajian Awal | 46 |
| 3.2.2 | Fasa II: Analisa Sistem | 46 |
| 3.2.2.1 | Analisa Sistem | 47 |
| 3.2.2.2 | Analisa Keperluan Sistem | 48 |
| 3.2.2.3 | Analisa Keperluan Perisian | 52 |
| 3.2.2.4 | Analisa Keperluan Perkakasan | 57 |
| 3.2.2.5 | Analisa Rekabentuk Antaramuka | 58 |

| | | |
|---------|---------------------------------------|----|
| 3.2.2.6 | Analisa Keupayaan Sistem | 58 |
| 3.2.3 | Fasa III: Rekabentuk Sistem | 59 |
| 3.2.4 | Fasa IV: Pengaturcaraan/Pengkodan | 60 |
| 3.2.5 | Fasa V: Pengujian dan Penyelenggaraan | 61 |

BAB 4: REKABENTUK SISTEM

| | | |
|-------|---------------------------|----|
| 4.1 | Pengenalan | 62 |
| 4.2 | Rekabentuk Struktur | 62 |
| 4.3 | Rekabentuk Proses | 63 |
| 4.4 | Rekabentuk Pangkalan Data | 65 |
| 4.4.1 | Rajah Hubungan Entiti | 66 |
| 4.4.2 | Jadual Pangkalan Data | 67 |
| 4.5 | Rekabentuk Antaramuka | 71 |

BAB 5: IMPLEMENTASI SISTEM

| | | |
|-------|--|----|
| 5.1 | Pendahuluan | 73 |
| 5.1.1 | Membina dan Menguji Pangkalan Data | 74 |
| 5.1.2 | Membina dan Menguji Program | 74 |
| 5.1.3 | Menginstall dan Menguji Sistem Baru | 75 |
| 5.1.4 | Menghantar Sistem Baru untuk Pengoperasian | 76 |
| 5.2 | Pembinaan Sistem Direktori Informasi Bas | 76 |
| 5.2.1 | Penyediaan Kandungan | 77 |

| | |
|--|--------|
| 5.2.2 Pengintegrasian Kandungan, Persembahan dan Pemrograman | 78 |
| 5.2.3 Fasa Pengkodan | 79 |
| 5.2.4 Kaedah Pengkodan | 80 |
| 5.2.5 Pendekatan Pengkodan | 81 |
| 5.3 Rumusan | 82 |
| BAB 6: PENGUJIAN SISTEM | 83 |
| 6.1 Pendahuluan | 83 |
| 6.2 Jenis-Jenis Pengujian | 85 |
| 6.2.1 Pengujian Modul dan Integrasi | 86 |
| 6.2.2 Pengujian Unit | 86 |
| 6.3 Jenis-Jenis Kesalahan | 87 |
| 6.3.1 Kesalahan Algoritma | 87 |
| 6.3.2 Kesalahan Sistem | 88 |
| 6.3.3 Kesalahan Dokumentasi | 88 |
| 6.4 Pengujian Sistem | 89 |
| 6.4.1 Ujian Unit | 90 |
| 6.4.2 Ujian Integrasi | 91 |
| 6.4.3 Ujian Fungsi | 92 |
| 6.4.4 Ujian Prestasi | 93 |

| | |
|--|--------|
| 6.4.5 Ujian Penerimaan | 93 |
| 6.4.6 Ujian Pemasangan | 93 |
| 6.5 Teknik Pengujian Sistem | 94 |
| 6.6 Menghasilkan Fail <i>Executable</i> | 96 |
| 6.7 Kesimpulan | 96 |
| BAB 7: PENYELENGGARAAN DAN PENILAIAN SISTEM | 97 |
| 7.1 Penyelenggaraan Sistem | 97 |
| 7.1.1 Keperluan Penyelenggaraan | 97 |
| 7.1.2 Metodologi Penyelenggaraan Sistem | 98 |
| 7.1.3 Pelan Pengembalian Bencana | 99 |
| 7.2 Penilaian Sistem | 100 |
| 7.2.1 Pendahuluan | 100 |
| 7.2.2 Pencapaian Objektif | 100 |
| 7.2.3 Masalah dan Penyelesaian | 102 |
| 7.2.3.1 Masalah Maklumat Laluan Bas Terkini | 102 |
| 7.2.3.2 Masalah Pangkalan Data | 103 |
| 7.2.3.3 Masalah Dalam Rekabentuk Sistem | 106 |
| 7.2.4 Maklumbalas Pengguna | 108 |
| 7.2.5 Kekuatan Sistem | 110 |
| 7.2.5.1 Lebih Efektif | 110 |
| 7.2.5.2 Ciri Keselamatan | 110 |

| | |
|--------------------------------------|-----|
| 7.2.5.3 Mesra Pengguna | 111 |
| 7.2.5.4 Lebih Fleksibel | 112 |
| 7.2.5.5 Trend Semasa | 112 |
| 7.2.6 Kekangan Sistem | 113 |
| 7.2.6.1 Sistem Pengemaskinian Sistem | 113 |
| 7.2.6.2 Pengujian Penerimaan | 113 |
| 7.2.7 Pembaharuan Masa Hadapan | 114 |
| 7.3 Rumusan | 115 |
| KESIMPULAN | 116 |
| RUJUKAN | 117 |
| LAMPIRAN | 119 |

SENARAI RAJAH

| NAMA RAJAH | HALAMAN |
|---|---------|
| Rajah 1.1: Jadual Perancangan Pembangunan Projek | 9 |
| Rajah 2.1: Algoritma <i>Generate and test</i> | 17 |
| Rajah 2.2: Graf mudah | 25 |
| Rajah 2.3: Jadual hasil pengiraan algoritma bagi Graf mudah | 25 |
| Rajah 2.4: Algoritma Dijkstra diaplikasikan pada peta | 27 |
| Rajah 2.5: Jadual Algoritma bagi peta | 28 |
| Rajah 2.6: Antaramuka paparan bagi <i>Bus Information System</i> | 33 |
| Rajah 2.7: Struktur ringkas <i>BIS</i> | 34 |
| Rajah 2.8: Antaramuka <i>Bus Information System</i> | 35 |
| Rajah 2.9: Antaramuka bagi <i>Madrid Region Government Transport Information System</i> | 38 |
| Rajah 2.10: Antaramuka pilihan lokasi/destinasi | 39 |
| Rajah 3.1: Fasa Pembangunan Projek | 45 |
| Rajah 4.1: Carta Struktur Sistem Informasi Direktori Bas | 63 |
| Rajah 4.2: Gambarajah Konteks | 64 |
| Rajah 4.3: Aliran Data Proses Perancangan Laluan | 65 |
| Rajah 4.4: Hubungan Entiti (ER) | 66 |
| Rajah 4.5: Contoh unsur NODE dalam XML | 68 |

| | |
|---|-----|
| Rajah 4.6: Contoh unsur LINKS dalam XML | 70 |
| Rajah 4.7: Contoh Antaramuka Form Pilihan & Paparan | 72 |
| Rajah 6.1: Struktur Pengujian Sistem | 90 |
| Rajah 6.2: Ujian Unit | 95 |
| Rajah 7.1: Nama Baru Sistem | 101 |
| Rajah 7.2: Paparan XML pada Pelayar | 105 |
| Rajah 7.3: Contoh Dokumen XML | 106 |

BAB 1

PENGENALAN

1.1 PENGENALAN PROJEK

Sebagaimana yang diketahui, teknologi komputer telah lama wujud dan digunakan dalam kehidupan seharian kita. Pada awalnya, komputer agak perlahan, mempunyai saiz yang besar (boleh merangkumi satu bilik), berat dan kos operasinya adalah mahal. Jadi penggunaannya adalah terhad. Komputer hanya digunakan dalam keadaan penting sahaja, sebagai contoh dalam bidang sains dan peperangan. Ia hanyalah digunakan oleh golongan yang berkenaan dan mampu sahaja. Tetapi dengan berkembangnya teknologi komputer dan bahasa pengaturcaraan, secara tidak langsung ia memberi kesan yang mendalam dalam aspek kehidupan manusia. Selain memudahkan tugas manusia, ia juga mempercepatkan segala urusan. Dalam zaman teknologi yang maju ini, segala pekerjaan perlu dilakukan dengan pantas. Pelbagai organisasi dan industri kini bergantung kepada teknologi komputer dalam menjalankan aktiviti-aktiviti dan operasi harian mereka. Ini termasuklah operasi dalam industri pengangkutan. Penggunaan komputer dalam industri pengangkutan tidak dapat disangkal lagi dengan wujudnya komputer-komputer yang memainkan peranan penting dalam ketepatan dan keselamatan perjalanan sesebuah medium transportasi. Sebagai contoh, LRT (Light Rail Transit) bergantung sepenuhnya kepada komputer untuk menjalankan operasi harian. Bagi pihak pengguna pula terdapat pelbagai kemudahan berkomputer yang dapat memudahkan perjalanan mereka. Sebagai contoh, sistem tiket berkomputer dan lain-lain. Salah satu kemudahan yang agak penting bagi pengguna kemudahan pengangkutan hari ini ialah sistem informasi

berkomputer. Sistem informasi ini membekalkan pengguna dengan pelbagai maklumat penting seperti masa perjalanan, tempat berlepas dan lain-lain lagi. Tanpa sesebuah sistem informasi yang sistematik dan lengkap, pengguna akan sedikit sebanyak kehilangan salah satu kemudahan yang sepatutnya memberi kesenangan kepada mereka. Sewajar dengan ini, saya telah bercadang untuk membangunkan sebuah sistem informasi direktori bas. Sistem informasi ini kelak akan membekalkan pengguna bas dengan pelbagai maklumat penting terutamanya mengenai perjalanan bas, masa bas tersebut berlepas, jumlah tambang yang akan dikenakan. Pengguna juga dapat merancang perjalanan mereka dengan menggunakan sistem ini, sebagai contoh, jika pengguna telah memasukkan lokasi berlepas dan destinasi yang di tuju, sistem akan memaparkan bas-bas yang perlu diambil (jika perlu), dimana bas-bas tersebut perlu diambil, anggaran masa perjalanan dan akhirnya jumlah tambang yang akan dikenakan. Ternyata disini sistem ini amat berguna kepada pengguna terutamanya bagi pengguna baru yang menggunakan perkhidmatan bas.

1.2 MASALAH YANG WUJUD

Pelbagai masalah telah dikenalpasti yang menyokong kepada pembangunan sistem.

Masalah-masalah ini ialah: -

- Pengguna bas ingin merancang perjalanan mereka terlebih dahulu sebelum mereka menaiki bas. Ini memudahkan mereka kerana setelah merancang mereka tahu bas apa yang perlu dinaiki, dimana harus mereka ambil dan turun dan berapa yang perlu dibayar. Secara tak langsung, ia menjimatkan masa.
- Pengguna bas inginkan suatu sistem yang boleh dipercayai dan mudah. Biasanya kebanyakan mereka terpaksa bertanya kepada pengguna lain, informasi yang dibekalkan adakalanya tidak sahih.
- Mereka ingin merancang dan memperolehi maklumat dengan selesa. Mereka tidak perlu beratur atau bersusah payah untuk bertanya atau mencari papan tanda direktori lebih-lebih lagi di waktu sibuk.
- Mereka inginkan suatu sistem yang dapat menyediakan informasi dengan cepat dan sistematik. Dengan menggunakan komputer, pengguna dapat memperolehi maklumat yang lengkap dan kini tanpa membuang masa.
- Pengguna mahukan informasi yang sentiasa ada. Kebanyakan pengguna mengetahui bahawa terdapat papan tanda informasi yang tidak dijaga dengan baik. Maklumat yang terkandung dalam papan tanda ini juga kadang-kadang tidak dikemaskinikan. Buku panduan yang disediakan pihak bas sering kehabisan dan sukar untuk diperolehi kerana ianya hanya ada di tempat tertentu.

1.3 MASALAH YANG DIHADAPI

Terdapat beberapa masalah yang wujud semasa membuat analisa mengenai tajuk:

- Maklumat mengenai direktori bas yang sedia ada tidak beberapa lengkap dan sistematik.
- Sukar untuk mencari direktori bas yang lengkap terutamanya bagi pengguna bas sendiri.
- Maklumat direktori yang tidak kemaskini.
- Masalah juga hadir dari bas itu sendiri dimana mungkin bas akan mengalami kelewatan yang disebabkan oleh pelbagai faktor (proses kajian pemerhatian).
- Pengguna bas tidak beberapa minat untuk mengisi borang soal-selidik.
- Staf syarikat bas yang kurang memberi kerjasama.

1.4 OBJEKTIF

Objektif untuk projek ini adalah seperti berikut:

- Menghasilkan suatu direktori tanpa menggunakan kertas (paperless).
- Menghasilkan suatu sistem informasi yang dapat menyalur dan memproses maklumat berkaitan dengan waktu dan perjalanan bas.
- Menggabungkan kesemua direktori-direktori perjalanan bas kedalam suatu direktori asas yang boleh diakses secara online.
- Memudahkan pengguna-pengguna bas untuk merancang perjalanan mereka.
- Memperkenalkan satu kaedah persembahan maklumat rekod yang lebih sistematik dengan penggunaan komputer.

1.5 SKOP PROJEK

Sistem yang dibangunkan ini bertujuan untuk menyediakan direktori bas khususnya kepada pengguna-pengguna bas secara online. Skop pembangunan ini dibahagikan kepada :

Ciri-ciri sistem informasi

Sistem ini ialah sistem atas talian (online) dimana ianya boleh diakses dimana-mana jua dengan menggunakan komputer.

Sistem juga menyediakan modul bagi memudahkan pengguna merancang perjalanan mereka dimana sistem boleh mengesyorkan bas atau bas-bas yang perlu dinaiki serta jumlah tambang dan masa yang diperlukan untuk tiba di destinasi yang di pilih.

Secara amnya, sistem akan melakukan perkara yang disebut dibawah: -

- ❑ Menerima input daripada pengguna.
- ❑ Merancang perjalanan bas terpendek berdasarkan kepada destinasi pengguna.
- ❑ Memaparkan hasil pengiraan laluan.

Bahasa

Bahasa yang akan digunakan di dalam pembangunan sistem informasi direktori bas ini adalah bahasa melayu (bahasa ibunda) yang mudah difahami kerana sistem ini disasarkan kepada pengguna-pengguna yang terdiri daripada pelbagai bangsa di Malaysia.

Kawasan Laluan

Kawasan laluan bas yang diliputi dalam sistem ialah seluruh laluan bas di Kuala Lumpur dan Selangor. Kawasan Laluan yang dipilih ini ialah kawasan laluan bas Intrakota Sdn. Bhd. Syarikat bas ini menyediakan perkhidmatan bas penumpang yang paling baik dan sistematik di Malaysia.

1.6 SASARAN PENGGUNA

Pengguna-pengguna perkhidmatan bas

Sistem Informasi Direktori Bas ini ditujukan terutamanya kepada golongan pengguna perkhidmatan bas yang baru dikawasan Kuala Lumpur/Selangor. Pengguna bas dapat memperoleh pelbagai maklumat penting mengenai bas yang bakal dinaiki. Sebagai contoh, pengguna boleh memperolehi maklumat mengenai tambang serta laluan bas. Mereka juga dapat merancang perjalanan mereka dengan hanya memberitahu sistem destinasi yang ingin dicapai. Sistem kemudiannya akan menyatakan bas mana yang perlu dinaiki dan jika terdapat sebarang pertukaran bas yang perlu dilakukan, sistem juga akan syorkan bas mana yang perlu dinaiki.

1.7 PERANCANGAN PROJEK

Projek ini dimulakan pada semester tiga sesi 2001/2002. Tarikh pengesahan tajuk ialah pada 18 Mac 2002. Projek ini dibahagikan kepada dua fasa dimana fasa satu akan dilaksanakan pada semester tiga sesi 2001/2002 dan fasa dua dilaksanakan pada semester satu sesi 2002/2003. Fasa satu melibatkan kajian Awal, Analisa Sistem dan Rekabentuk Sistem. Analisa Sistem dimulakan pada 20 Mac 2002 dan akan berakhir pada 3 Mei 2001. Rekabentuk Sistem pula dijalankan pada 29 Mac 2002 dan akan berakhir pada 5 Jun 2000.

Fasa dua melibatkan proses pengkodan dan pengujian. Proses pengkodan dilaksanakan pada 5 Jun 2002 dan berakhir pada Oktober 2002. Pengujian dan perlaksanaan sistem dijalankan pada 5 Jun 2002 dan berakhir pada Oktober 2002. Proses dokumentasi dilakukan sepanjang perjalanan projek. Rajah 1.1 menunjukkan plan masa dengan menggunakan Carta Ghannt. Fasa-Fasa yang terlibat dalam pembangunan sistem ini :

Kajian Awal dan Analisa Sistem

- Menentukan objektif sistem
- Menentukan skop sistem
- Menentukan keperluan sistem
- Menyediakan skedul projek
- Memilih dan menentukan model pembangunan sistem

Rekabentuk Sistem

- Rekabentuk antaramuka sistem

Perlaksanaan / Pengkodan

- Mempelajari penggunaan membuat pengekodan Visual Basic
- Mempelajari penggunaan membuat pengekodan Visual InterDev

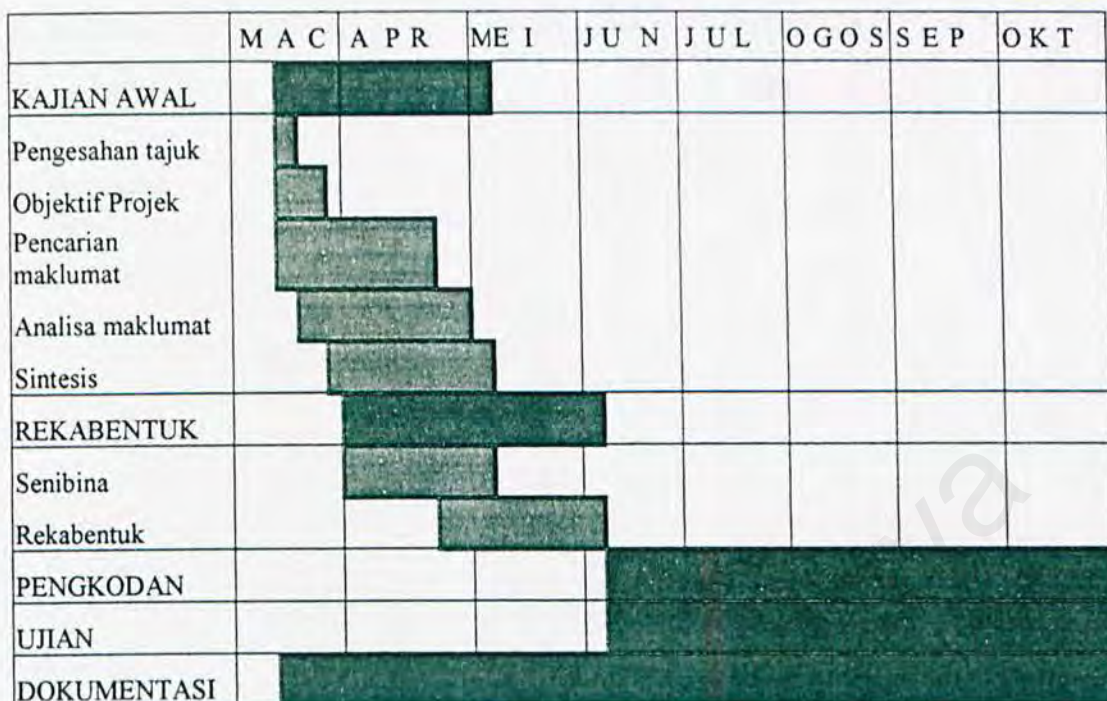
(Pembelajaran dilakukan melalui contoh yang sedia ada, buku, buku elektronik, internet (MSDN, dsb)).

Pengujian Sistem

- Rekabentuk data ujian
- Menguji modul-modul
- Membandingkan keputusan ujian dengan keputusan yang dijangkakan

Penyelenggaraan Sistem

- Memperbaiki dan melakukan pengemaskinian pada sistem



Rajah 1.1: Jadual Perancangan Pembangunan Projek

1.8 HASIL YANG DIJANGKAKAN

Daripada apa yang dijangkakan, sistem ini kelak dapat membantu pengguna dalam memperolehi maklumat secara lebih sistematik. Dengan menggunakan sistem informasi ini juga, ianya sedikit sebanyak dapat menjimatkan usaha-usaha pihak-pihak tertentu untuk menyediakan suatu kawasan strategik dimana direktori boleh diletakkan. Dengan ini, tidak perlulah untuk pengguna ke kawasan yang tertentu bagi melihat direktori tersebut. Mereka hanya perlu mempunyai akses kepada sebuah komputer dan talian internet untuk memperoleh direktori disamping perkhidmatan lain yang disediakan oleh sistem. Perkhidmatan lain ini termasuklah perancangan perjalanan. Sistem ini juga direka supaya ianya mudah digunakan oleh

pengguna. Pengguna hanya perlu mengetahui lokasi berlepas dan destinasi yang ingin dituju dan sistem akan membuat perancangan untuk mereka. Perancangan ini adalah dari segi laluan terpantas/terpendek yang boleh diambil pengguna untuk tiba ke destinasi.

University of Malaya

BAB 2

KAJIAN LITERASI

BAHAGIAN I: KAJIAN DAN PENULISAN

2.1 TUJUAN

Kajian literasi ialah kajian yang dilakukan secara am keatas sistem yang akan dibangunkan. Ia dilakukan untuk beberapa tujuan utama, antaranya ialah :

- Untuk mengumpul maklumat berkenaan sistem yang akan dibangunkan.
- Untuk mengkaji dan menilai sistem yang mempunyai konsep yang sama atau releven, yang telah dibangunkan bagi menentukan kelemahan dan kekuatan sistem tersebut disamping memperbaiki kelemahan sistem yang telah dikenalpasti.
- Untuk mendapatkan pemahaman yang jelas tentang konsep yang terlibat di dalam sistem yang bakal dibangunkan disamping membandingkan beberapa perisian yang akan digunakan bagi mendapatkan hasil dan penyelesaian yang terbaik.

2.2 DEFINISI SISTEM

Perkataan sistem mempunyai beberapa makna iaitu (1); cara atau kaedah untuk melakukan sesuatu, (2); kumpulan beberapa bahagian yang sama-sama bekerja untuk satu tujuan dan (3); kumpulan pendapat yang teratur untuk melakukan sesuatu.

2.2.1 Definisi maklumat

Perkataan maklumat berasal daripada beberapa perkataan yang bermaksud makluman, pemberitahuan, keterangan dan pengetahuan.

2.2.2 Definisi Direktori

Direktori ialah suatu kumpulan data yang teratur. Data ini biasanya mengandungi entiti (cth: Nama) dan atribut yang menerangkannya (cth: No. Telefon).

2.3 PENEMUAN RUJUKAN / KAEDAH KAJIAN

Secara amnya, pembangunan sistem tidak akan sempurna jika tiada pengumpulan dan penyelidikan maklumat mengenai sistem yang bakal dibangunkan. Oleh itu, maklumat yang dikumpul amat penting bagi menentukan sistem mencapai matlamat dan objektif dengan tepat. Maklumat boleh diperolehi daripada pelbagai sumber dan setiap sumber memberi maklumat yang berbeza serta memerlukan teknik pencarian yang berbeza. Beberapa kaedah telah digunakan untuk menjalankan kajian serta analisa terhadap system sedia ada dan sistem yang bakal direkabentuk. Di antaranya ialah kaedah pengumpulan data dan kaedah penulisan.

2.3.1.1 KAEDAH PENGUMPULAN DATA

Temubual atau temuramah

Kaedah ini telah dipilih untuk memperoleh maklumat mengenai sistem informasi bus yang sedia ada. Temubual dan temuramah secara tidak formal telah diadakan terutamanya di perhentian dan stesyen bas. Temubual ini dijalankan pada pengguna bas dan pekerja syarikat bas untuk memperoleh maklumat secara kasar. Temubual juga telah dijalankan pada rakan-rakan serta ahli keluarga dari semasa ke semasa. Maklumat yang diperolehi, secara tidak langsung telah dapat memberi pelbagai pandangan yang berbeza terhadap domain. Pandangan yang diberikan ini membantu dalam pemahaman masalah yang wujud dengan sistem yang ada dan sistem yang akan dibangunkan.

Pemerhatian

Pemerhatian juga telah dilakukan di stesyen dan perhentian bas. Pemerhatian ini dijalankan pada masa-masa tertentu seperti pada waktu sibuk dan waktu lengang. Tujuan pemerhatian dijalankan adalah untuk melihat kekerapan penggunaan bas pada waktu tersebut. Semasa pemerhatian, ada segelintir pengguna yang didapati tidak beberapa biasa dengan penggunaan bas, kebanyakan mereka kelihatan menanya pengguna-pengguna lain untuk maklumat dan tunjuk ajar. Di sesetengah perhentian bas, maklumat mengenai bas yang akan berhenti tidak wujud langsung, ini menyebabkan mereka terpaksa bertanya kepada pengguna lain dan pemandu bas itu sendiri.

Pemerhatian bagi sistem informasi bas yang terdapat di Malaysia tidak dilakukan kerana buat masa ini sistem sebegini masih tidak wujud. Informasi mengenai sistem informasi bas di Malaysia telah diperolehi melalui internet dan pertanyaan. Syarikat bas yang sedia ada hanya menyediakan on-line directory/laman web syarikat dimana semua informasi mengenai jadual perjalanan, syarikat, dan tambang diletakkan.

Dokumentasi / Buku dan Majalah

Kajian dan analisa telah dibuat terhadap dokumen dan bahan penulisan yang mempunyai hubungan dengan sistem yang bakal direkabentuk. Pengumpulan-pengumpulan maklumat dari buku dan majalah juga dilakukan untuk menambahkan lagi hasil kajian.

Dokumen-dokumen tersebut diperolehi melalui rujukan di perpustakaan UM, melalui koleksi peribadi serta diperolehi dari rakan-rakan.

Melayari Internet

Internet telah banyak membantu dalam pembangunan sistem informasi bas ini. Dengan melayari Internet serta *on-line databases*, pelbagai maklumat telah diperolehi bagi tujuan membuat kajian terhadap sistem yang telah wujud untuk dibuat perbandingan.

2.3.2 KAEDAH PENULISAN

Analisa Bahan Penulisan

Analisa telah dibuat terhadap data yang diperolehi dengan menghurai semula data dan maklumat tersebut kepada bentuk yang lebih ringkas supaya mudah difahami serta menepati objektif pembangunan projek.

Komparatif

Kesimpulan dan keputusan telah dibuat melalui perbandingan dengan sistem yang telah wujud dan sistem yang bakal dibangunkan berpandukan data yang diperolehi.

2.4 KAJIAN BERKAITAN ALGORITMA SHORTEST PATH

2.4.1 HEURISTIC SEARCH

Heuristic Search ialah suatu metod/cara pencarian graf yang mana:-

- Ia tidak semestinya selalu mencari penyelesaian yang terbaik.
- Tetapi ia dijamin dapat mencari penyelesaian dalam masa yang munasabah.
- Dengan mengurangkan pencarian secara menyeluruh ia menjadi lebih efisien.
- Berguna dalam menyelesaikan masalah yang susah dimana:-
 - Ia tidak boleh diselesaikan dengan cara yang lain.
 - Penyelesaian masalah mengambil masa yang lama atau tak terhingga untuk diselesaikan.

Suatu contoh klasik metod/cara *heuristic search* ialah masalah *travelling salesman*.

Antara contoh algoritma *heuristic search* ialah Generate and test dan Best-first Search/A*.

- **Generate and test**

Generate and test secara amnya ialah pencarian *depth first* dimana suatu penyelesaian yang lengkap mesti dicipta sebelum ianya boleh diperiksa. Ia berfungsi dengan mewujudkan penyelesaian secara rawak. Penyelesaian secara rawak ini tidak semestinya bermaksud penyelesaian kepada masalah tersebut akan dijumpai. Dalam bentuk ini, ia lebih dikenali sebagai algoritma *British Museum*, berdasarkan kepada cara yang digunakan untuk mencari sesuatu objek didalam British Museum dengan berjalan secara rawak!. Secara ringkasnya, algoritma *British Museum* ini adalah praktikal hanya bila menyelesaikan masalah yang mempunyai *search tree* (pepohon gelintar) yang kecil. Apabila *search tree* terlalu besar, algoritma mungkin tidak akan menjumpai penyelesaian.

- Wujudkan penyelesaian yang mungkin. Penyelesaian bermaksud mewujudkan suatu laluan daripada *start state*. Untuk sesetengah masalah, ini bermaksud mewujudkan suatu titik dalam *problem space*.
- Periksa penyelesaian dengan membandingkan titik pilihan atau titik akhir untuk laluan pilihan dengan *goal states*.
- Jika penyelesaian telah dijumpai, keluar dari algoritma. Jika tidak, ulang langkah pertama.

Rajah 2.1: Algoritma *Generate and test*

- **Best-first Search/A***

Algoritma A* mungkin adalah algoritma pencarian yang paling sesuai kerana ia akan mencari laluan terpendek dengan kos yang paling kurang. Algoritma ini adalah hasil gabungan algoritma *Greedy Search* dan algoritma *Uniform Cost Search*. Algoritma *Greedy Search* akan memberikan laluan dalam masa yang terpendek ($h(n)$) manakala algoritma *Uniform Cost Search* akan memberikan laluan dengan kos yang paling minimal ($g(n)$). Dengan menggabungkan kedua-dua algoritma ini, algoritma yang diperolehi ialah:-

$$f(n) = h(n) + g(n)$$

Untuk memahami dengan lebih lanjut bagaimana algoritma A* berfungsi, kita mesti mengkaji algoritma-algoritma yang membentuk algoritma A*.

Algoritma Greedy Search

Algoritma *Greedy Search* akan mencari laluan terpendek dengan menggunakan fungsi heuristik. Fungsi ini akan mengurangkan masa pencarian, tetapi, ia mungkin tidak akan menjumpai laluan yang paling optimal (laluan paling murah). Algoritma ini mencari laluan ke *goal state* hanya daripada nod semasa dan tidak mengambil kira laluan yang lain yang mungkin wujud sepanjang laluan yang diambil. Oleh sebab algoritma ini tamak (*greedy*) ia akan mencari laluan ke *goal state* dengan pantas. Adakalanya algoritma *greedy* ini tidak berfungsi dengan betul/tepat kerana ia mungkin akan mengembangkan nod-nod yang akan membawa kepada jalan buntu. Algoritma ini akan mengikut satu laluan berterusan sehingga ia mencapai *goal state*, yang menjadikannya

pantas, tetapi berkemungkinan menyebabkan ia tidak memperoleh sebarang hasil. Suatu contoh bagaimana algoritma ini berfungsi ditunjukkan dibawah. Kita akan menggunakan fungsi heuristik: -

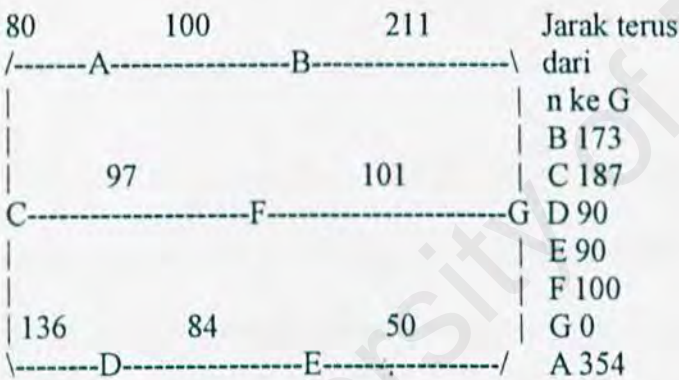
$h(n)$ = jarak antara n dan goal state

| | | | |
|---------------------|-----|-----|-------------|
| 80 | 100 | 211 | Jarak terus |
| /-----A-----B-----\ | | | dari |
| | | | n ke G |
| | | | B 173 |
| | 97 | 101 | C 187 |
| C-----F-----G | | | D 90 |
| | | | E 90 |
| | | | F 100 |
| 136 | 84 | 50 | G 0 |
| \-----D-----E-----/ | | | A 354 |

Kita akan mula dengan A dan cari laluan ke G. Nod B dan C akan dikembangkan dari A. Di C, jarak ke G ialah 187 dan dari B ialah 173. Memandangkan jarak lurus ke B lebih pantas (laluan ke G melalui B hanya mengambil 2 nod: B dan G; lebih pantas) , kita akan mengambil laluan melalui B. Algoritma ini mengambil laluan yang terpanjang daripada posisi semasa. Nod yang seterusnya ialah G, iaitu destinasi kita. Sebagai mana yang boleh dilihat, algoritma ini menemui laluan dengan pantas, tetapi, ianya bukan yang paling murah. Keseluruhan jarak melalui A-B-G ialah $100+211 = 311$. Laluan yang lebih murah boleh diperolehi dengan melalui A-C-F-G dan jaraknya ialah $80 + 97 + 101 = 278$.

Uniform Search Algorithm

Algoritma ini akan mencari laluan paling optimal dengan mengurangkan kos laluan semasa. Algoritma akan mengembangkan semua nod yang mungkin daripada nod awalan. Ia akan terus mengembangkan semua nod daripada nod awalan sehingga laluan termurah dicapai, bermaksud walaupun *goal state* telah dicapai, algoritma akan terus mencari laluan yang lain selagi ianya lebih murah. Jika *goal state* telah dicapai dengan laluan yang lebih murah daripada laluan alternatif yang lain, algoritma akan berhenti tanpa mengkaji laluan yang selebihnya. Algoritma Uniform ini menggunakan fungsi $g(n) = DEPTH(n)$.



Sebagai contoh, kita akan menggunakan rajah yang sama seperti diatas. Bermula di A dan berakhir di G. Nod B dan C akan dikembangkan dari A. Kemudian D dan F akan dikembangkan dari C. Sekarang, kita akan kembangkan nod G dari B. Kita telah mencapai goal dengan jarak 311. Laluan ini adalah lebih panjang daripada laluan alternatif (A-C-D = 216; A-C-F = 177), oleh itu kita akan terus mencari laluan termurah (terpendek). Daripada D, kita akan ambil laluan melalui E dan ini akan memberi kita jarak keseluruhan sebanyak 319 (216+84). Sekarang kita akan cuba laluan melalui F ke

G dan ia akan membawa kita kepada goal dengan jarak sebanyak 278. Kita akan berhenti disini, memandangkan jaraknya adalah lebih pendek daripada dua laluan yang lain.

Sekarang dengan pengetahuan mengenai kedua-dua algoritma yang diterangkan diatas, cara algoritma A* berfungsi lebih mudah difahami. Algoritma Greedy menjadikan pencarian pantas manakala algoritma Uniform akan memberi laluan termurah. Fungsi algoritma A* ditunjukkan dibawah: -

$g(n)$ = laluan termurah dari permulaan ke nod n

$h(n)$ = laluan termurah dari n ke goal

Oleh itu, $f(n)$ ialah laluan termurah melalui n

Algoritma A* menetapkan bahawa fungsi heuristik $h(n)$ supaya *underestimate* kos untuk mencapai goal. Dengan menggunakan teknik *underestimate* ia akan mencari laluan yang munasabah dalam masa yang singkat. Kita akan melakukan ujian kepada algoritma A* dan membandingkannya dengan algoritma-algoritma sebelum ini.

$$F(n) = g(n) + h(n)$$

$H(n)$ = Laluan terus antara n dan goal state

$G(n)$ = DEPTH(n)

| | | | |
|---------------------|-----|-----|-------------|
| 80 | 100 | 211 | Jarak terus |
| /-----A-----B-----\ | | | dari n ke G |
| | | | A 250 |
| | | | B 173 |
| | 97 | 101 | C 187 |
| C-----F-----G | | | D 90 |
| | | | E 90 |
| | | | F 100 |
| 136 | 84 | 50 | G 0 |
| \-----D-----E-----/ | | | |

Kita akan mula dari A ke G. Nod B dan C dikembangkan dari A. Laluan A-B akan memberi kos: $100 (g(n)) + 173 (h(n)) = 273$. Laluan lain yang melalui A-C akan memberi kos: $80 (g(n)) + 187(h(n)) = 267$. Disini kita akan membezakan algoritma A* daripada dua algoritma sebelum ini. Algoritma Greedy tidak akan melihat kepada laluan A-C untuk mencari G dan algoritma Uniform akan mencari semua laluan yang mungkin. Algoritma A* tidak akan melihat laluan A-B kerana laluan A-C adalah lebih murah. Daripada C, nod F dan D dikembangkan. Laluan A-C-D mempunyai kos $(80 + 136) (g(n)) + 90 (h(n)) = 306$. Laluan A-C-F mempunyai kos $(80 + 97) (g(n)) + 100 (h(n)) = 277$. Laluan yang kedua ini lebih pendek, jadi algoritma akan terus melalui laluan A-C-F ke G. Ini adalah cara bagaimana algoritma A* berfungsi. Ianya agak cepat dan akan mencari laluan yang paling optimal.

Masalah yang tidak boleh diselesaikan

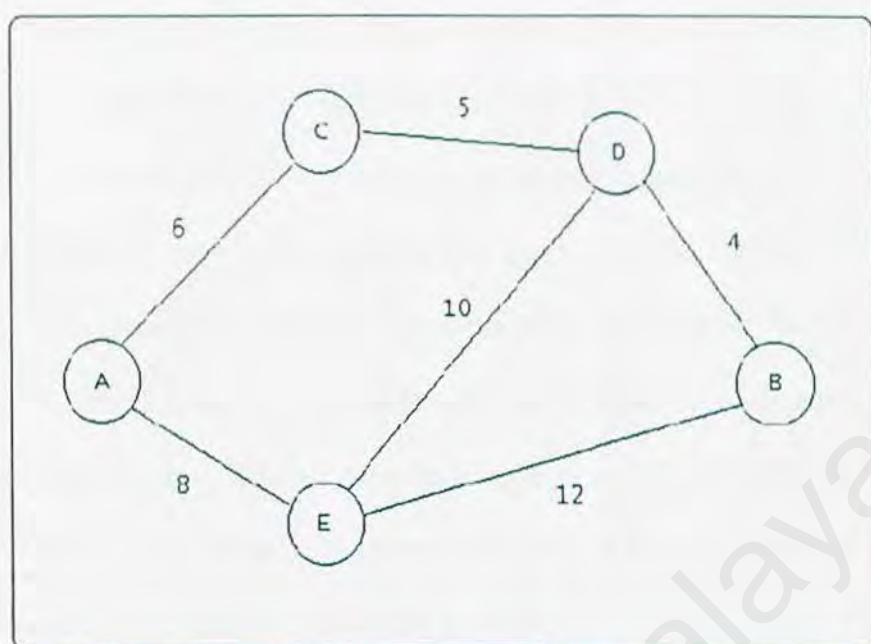
Walaupun algoritma A^* adalah efisien, tetapi ia tidak bermaksud ia boleh menyelesaikan semua masalah. Masalah yang tidak boleh diselesaikan termasuklah graf infinit dan graf yang terlalu besar. Penyelesaian bagi graf infinit mungkin tidak wujud, jadi ia tak menjadi masalah yang besar. Algoritma A^* menyimpan kesemua nod yang dikembangkan kedalam ingatan, jadi ia akan kehabisan ruang ingatan jika ingin menyelesaikan masalah yang besar.

2.4.2 ALGORITMA DIJKSTRA

Algoritma Dijkstra ialah suatu cara yang sistematik untuk mencari jalan yang paling cepat dalam suatu graf pemberat (weighted graph) dari titik A ke titik B. Algoritma ini telah diperkenalkan pada tahun 1959. Pemberat (weights) yang terikat kepada bucu (edges) digunakan untuk mewakili kuantiti seperti jarak, kos, dan masa. Secara am, jarak melalui suatu laluan ialah jumlah pemberat laluan itu.

Pertama, algoritma Dijkstra ini akan diterangkan secara terperinci dan kemudiannya contoh akan diberi berdasarkan peta kampus Penn State University Park.

- Cari atau cipta suatu peta dengan pelbagai tempat, atau nod dan jalan, atau bucu (edges) yang mengaitkan mereka.
- Letakkan nilai keatas setiap bucu (edges) yang mewakili jarak, masa, kos ataupun sebarang jenis pemberat.
- Cipta satu jadual dimana lajurnya mewakili setiap nod.
- Bilangan barisan bergantung kepada peta yang digunakan. Setiap barisan mewakili 1gelung (an iteration) algoritma. Dalam setiap barisan (mewakili 1 gelung), kita akan memasukkan jarak nod-nod dari titik permulaan yang tedapat dalam peta. Kita akan meletakkan * kepada masukan yang mana jaraknya adalah benar dan dipilih. Rajah dibawah menunjukkan suatu contoh mudah dan jadual berkaitan.



Rajah 2.2: Graf mudah

Jadual akhir bagi algoritma ditunjukkan dibawah. Hasil pengiraan algoritma berada di kanan jadual.

| NODES | Table of Values | | | | |
|-------|-----------------|----------------|-----------------|-----------------|-----------------|
| A | | | | | 0 ^u |
| C | 6 ^u | | | | 6 ^u |
| E | 8 | 8 ^u | | | 8 ^u |
| D | Unknown | 11 | 11 ^u | | 11 ^u |
| B | Unknown | Unknown | 20 | 15 ^u | 15 ^u |

Rajah 2.3: Jadual hasil pengiraan algoritma bagi Graf mudah

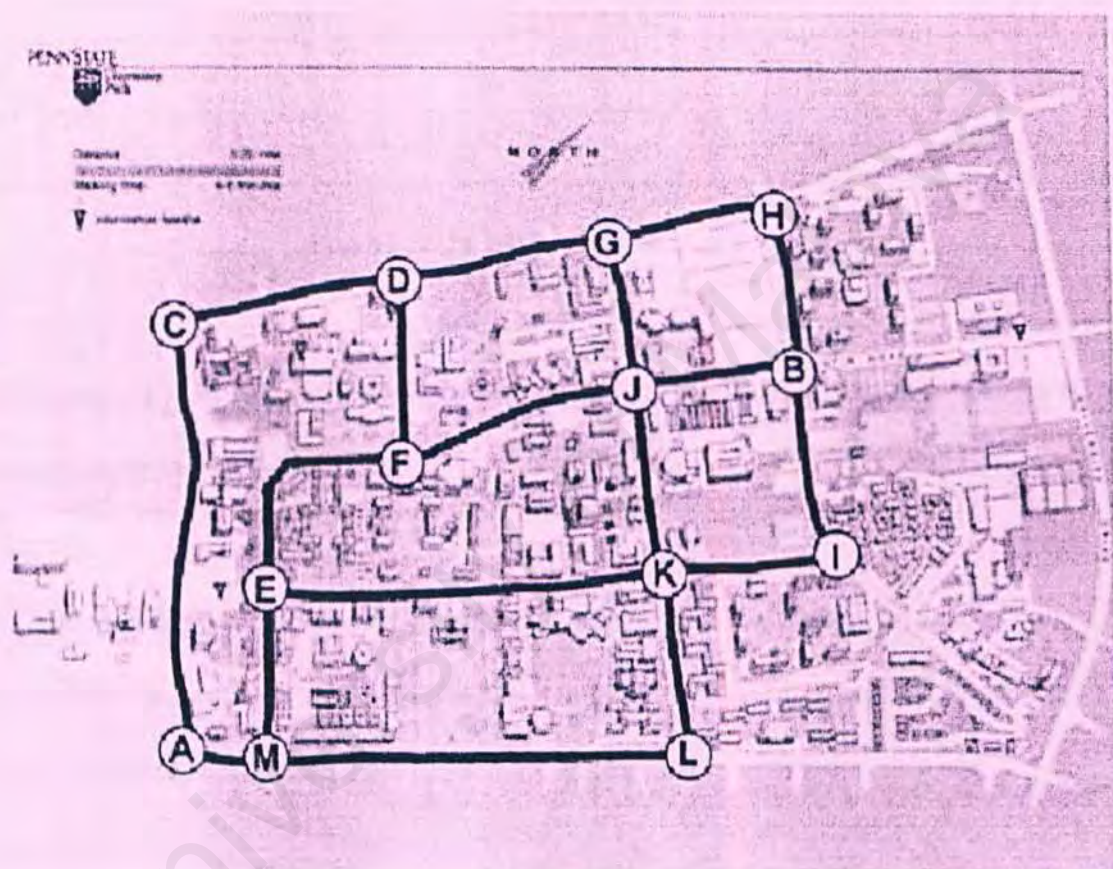
Di mukasurat sebelah, ditunjukkan bagaimana untuk mengisi jadual dengan menggunakan algoritma Dijkstra.

1. Pilih suatu titik permulaan dan cari semua jarak ke nod bersebelahan (nod yang bersambungan dengan jalan kepada nod/titik permulaan). Letakkan jarak-jarak

ini kedalam kotak yang bersesuaian dalam jadual. Jika suatu nod tidak dicapai, ia akan dilabel “unknown”, atau dibiarkan kosong. Sebagai contoh, dalam peta diatas, titik/nod permulaan ialah A dan nod bersebelahan ialah C dan E.

2. Letakkan bintang (*) pada nilai yang paling rendah dalam barisan. Nod dengan star dalam lajurnya dipanggil *settled*. Nilai jarak yang terakhir dan terpendek bagi nod tersebut dari titik/nod permulaan adalah nilai yang terpapar. Titik/nod permulaan sentiasa mempunyai jarak 0. Sebagai contoh, pada peta diatas, kita memilih C untuk diletakkan * kerana jaraknya ialah 6. Jika dibandingkan dengan jalan yang melalui E (8), ianya lebih pendek.
3. Sekerang kita akan mengisi barisan-barisan seterusnya. Mula dengan mencari semua nod yang bersebelahan dengan nod yang dipilih (*) (*settled*) dalam barisan semasa. Dalam contoh ini, dalam barisan pertama, A dan C telah dipilih, jadi nod yang bersebelahan ialah D (bersebelahan dengan C) dan E (bersebelahan dengan A). Kira jarak nod ini dari titik/nod permulaan melalui nod yang telah dipilih (*settled node*) yang mana mereka bersebelahan dan letakkan nilai kedalam kotak yang sesuai dalam jadual. Sebagai contoh, jarak dari A ke C ialah 6 dan jarak dari C ke D ialah 5, jadi nilai 11 akan diletakkan untuk D dalam barisan kedua. Jika nod (*unsettled*) berada bersebelahan dengan lebih daripada satu nod dipilih (*settled*), jarak yang paling kurang dipilih untuk diletakkan ke dalam jadual.
4. Ulangi langkah 3 dan 4 sehingga semua nod telah *settled* (*). Dalam jadual yang lengkap, perhatikan bahawa nod B mempunyai nilai 20 dalam barisan yang ketiga, tetapi ia bertukar kepada 15 dalam kotak yang seterusnya (barisan

berikutnya). Ini kerana nilai 15 dijumpai selepas nilai 20, dan sejak 15 adalah kurang daripada 20, ia mengambil-alih tempat tersebut dalam jadual. Jadual tersebut menunjukkan nilai yang terpendek dari titik/nod A. Dari nod A ke B, jarak terdekat ialah 15.



Rajah 2.4: Algoritma Dijkstra diaplikasikan pada peta

Diatas ditunjukkan contoh penggunaan algoritma Dijkstra keatas peta Penn State campus. Pemberat pada bucu (edges) graf dalam contoh ini ialah jangkaan jarak menggunakan skala pada peta.

| | | | | | | | | | | | | | | | |
|---|-----|------|-----|------|------|------|-----|------|------|------|-------|------|-------|-------|-------|
| A | 0* | | | | | | | | | | | | | | .00* |
| B | | | | | | | | | 1.13 | 1.13 | 1.13 | 1.13 | 1.13* | 1.13* | |
| C | .56 | .56 | .56 | .56* | | | | | | | | | | | .56* |
| D | | | | .85 | .81 | .81 | .81 | .81* | | | | | | | .81* |
| E | | .31 | .31 | .31* | | | | | | | | | | | .31* |
| F | | | | .60 | .60* | | | | | | | | | | .60* |
| G | | | | | | | | 1.09 | 1.09 | 1.09 | 1.09* | | | | 1.09* |
| H | | | | | | | | | | | 1.30 | 1.30 | 1.30 | 1.30* | |
| I | | | | | | | | 1.01 | 1.01 | 1.01 | 1.01* | | | | 1.01* |
| J | | | | | .92 | .92 | .92 | .92 | .92* | | | | | | .92* |
| K | | | .8 | .8 | .8 | .8 | .8* | | | | | | | | .80* |
| L | | .63 | .63 | .63 | .63 | .63* | | | | | | | | | .63* |
| M | .10 | .10* | | | | | | | | | | | | | .10* |

Rajah 2.5: Jadual Algoritma bagi peta

Daripada apa yang dapat dilihat, laluan terpendek dari A ke B ialah 1.13 batu. Laluan ini secara terperinci : A-M-E-F-J-B.

Algoritma Dijkstra ini akan digunakan untuk mencari jalan terpendek dari suatu nod permulaan ke suatu nod destinasi dengan mengintegrasikan algoritma A*. Sebagaimana yang diketahui, Algoritma Dijkstra akan mencari jalan terpendek dari satu nod permulaan ke semua nod. Dengan adanya algoritma A*, jalan yang terpendek antara nod permulaan dengan nod destinasi yang dipilih dapat dikenalpasti.

2.5 KELEMAHAN SISTEM MANUAL

Sebelum adanya sistem informasi berkomputer, pengguna perkhidmatan bas terpaksa bersusah payah untuk mencari maklumat mengenai sesuatu laluan bas. Mereka mungkin terpaksa mencari papan tanda yang memaparkan laluan-laluan bas atau terpaksa menaiki bas tersebut sendiri untuk mengetahui laluan. Ada juga pengguna yang lebih gemar bertanya kepada pengguna lain atau pemandu bas tersebut. Maklumat yang diterima semasa bertanya adakalanya tidak begitu tepat dan sahih. Panduan melalui telefon juga wujud bagi membantu pengguna, tetapi ianya jarang digunakan kerana tidak mendapat sambutan yang meluas dan kadar panggilan yang agak mahal. Perkhidmatan yang disediakan melalui panduan telefon ini adakalanya kurang memuaskan. Ini disebabkan oleh pelbagai faktor. Sebagai contoh, telefonis yang menjawab panggilan mungkin tidak beberapa mahir dalam kemahiran sosial dan mengambil masa yang lama untuk menjawab soalan yang ditanya oleh pengguna. Telefonis mungkin terpaksa merujuk kepada carta laluan bas dan mencari destinasi yang dilalui oleh bas tersebut. Ini memakan masa yang agak lama. Ada juga syarikat bas yang menyediakan buku panduan pengguna. Buku panduan yang disediakan biasanya mengandungi maklumat yang lengkap. Ia boleh didapati di tempat-tempat yang dikhaskan. Pengguna perlu datang ke tempat-tempat khas ini untuk memperolehi buku panduan. Keburukan buku panduan yang disediakan ialah ianya sentiasa tidak mencukupi dan ia tidak memberi kesan yang meluas kepada pengguna. Maksud kesan yang meluas disini ialah sasaran buku panduan hanyalah kepada individu atau sekumpulan individu dan bukan untuk seluruh masyarakat, hanya kepada sesiapa yang

betul-betul minat menggunakan perkhidmatan bas. Maklumat yang terkandung dalam buku panduan dan carta perlu dikemaskini dengan perubahan masa. Dengan menggunakan bahan-bahan yang disebut diatas, mengemaskini maklumat/data adalah sukar dan memakan masa.

2.6 KELEBIHAN SISTEM INFORMASI ATAS-TALIAN

1. Maklumbalas yang cepat

Proses pengemaskinian dan pencarian maklumat pengguna dapat dilakukan dengan cepat.

2. Penyimpanan data yang tinggi

Sebuah komputer memiliki keupayaan yang tinggi untuk menyimpan data dan ini membolehkan data yang banyak disimpan tanpa menghadapi masalah seperti saiz data besar.

3. Antaramuka pengguna yang menarik

Sistem Informasi atas-talian mempunyai pelbagai alatan bagi menyediakan antaramuka yang menarik dan sistematik.

4. Memudahkan proses pengemaskinian data

Penggunaan komputer memudahkan pihak pentadbir mengemaskini sistem yang sedia ada. Masalah pertindihan data juga tidak akan berlaku kerana ia dapat dikesan dengan mudah.

5. Sistematik

Maklumat yang terkandung dalam sistem informasi adalah teratur dan memudahkan penyelenggaraan.

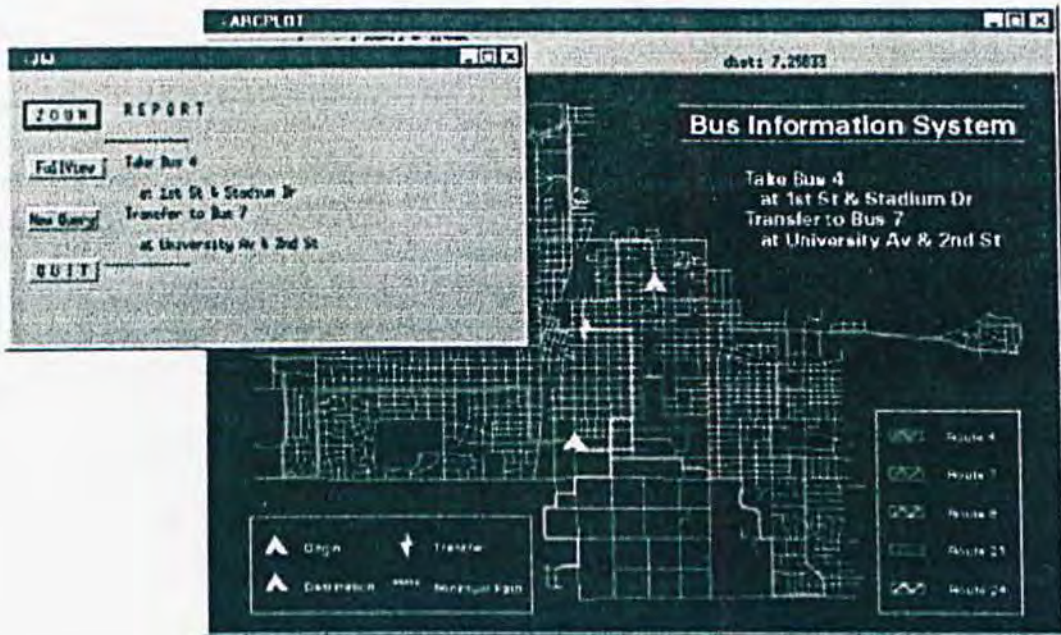
6. Mudah dicapai

Sistem Informasi atas-talian lebih senang dicapai oleh pengguna dimana-mana sahaja.

BAHAGIAN II: KAJIAN SISTEM SEDIA ADA

Setelah membuat kajian mengenai sistem secara meluas, terdapat sistem-sistem lain yang berkaitan dengan sistem informasi direktori bas yang dibangunkan oleh pihak lain. Pihak yang membangunkan sistem ini biasanya terdiri daripada pihak yang terlibat secara langsung dengan industri pengangkutan seperti pengusaha bas dan majlis pengangkutan di daerah tertentu. Sistem yang dibangunkan oleh mereka merangkumi kawasan laluan bas yang berlainan mengikut daerah atau laluan perkhidmatan bas mereka sendiri. Ciri-ciri bagi sistem informasi ini dihuraikan untuk dijadikan panduan dalam pembangunan sistem.

2.7 Champaign-Urbana Mass Transit District (CUMTD) BUS INFORMATION SYSTEM

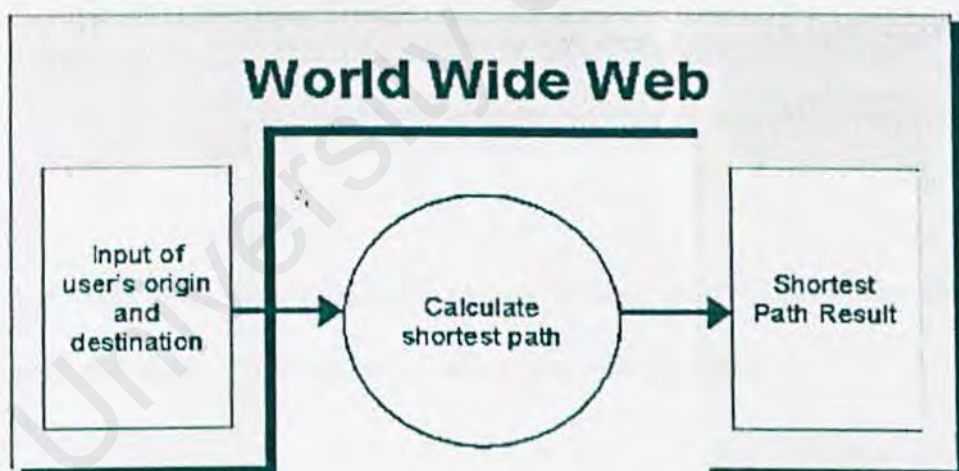


Rajah 2.6: Antaramuka paparan bagi *Bus Information System*

Bus Information System merupakan satu sistem *web-based* yang interaktif dan mudah digunakan. Ia dibangunkan oleh *Champaign-Urbana Mass Transit District (CU-MTD)*, sebuah kawasan yang merangkumi *University of Illinois*. *Bus Information System* juga wujud sebagai sistem *stand-alone* dikawasan-kawasan tertentu seperti di perpustakaan dan stesyen bas dimana penggunaannya adalah terhad. *Bus Information System* membolehkan pengguna memperoleh laluan bas yang terpendek setelah menginputkan lokasi berlepas dan destinasi yang di tuju. Lokasi berlepas dan destinasi yang dituju tidak semestinya menumpu ke perhentian bas sahaja, pengguna boleh menginputkan alamat sendiri. Alamat yang diinputkan mestilah merangkumi kawasan yang berada

dalam *BIS*. Apabila pengiraan laluan bas yang terpendek dilakukan, perhentian bas yang terdekat dengan alamat yang diberikan pengguna akan dirujuk. Antara kemudahan lain yang disediakan oleh *BIS* ialah maklumat mengenai laluan dan skedul bas. Dengan adanya *BIS*, pengguna terutamanya pengguna baru, dapat memperolehi maklumat mengenai bas dengan mudah dan sistematik. *BIS* dibangunkan dengan menggunakan *MapObject*, *NetEngine* dan *MapObject Internet Map Server* yang dibekalkan oleh *ESRI*. Struktur *BIS* terbahagi kepada tiga komponen utama iaitu:-

- input maklumat pengguna
- pengiraan laluan terpendek
- paparan hasil



Rajah 2.7: Struktur ringkas *BIS*



Step 1: Setting Time & Selection Method

Please provide the following information:

What Time Will Your Trip Begin?

: ☐ AM ☐ PM

- Provide Hour : Minute
- Example: 12:30

What Day of the Week is Your Trip?

☐ Weekday ☐ Saturday ☐ Sunday

Select Your Starting Point By:

- ☐ Address
- ☐ Landmark
- ☐ Intersection
- ☐ Point & click

- Selections made by Address or Landmark are approximate and may not reflect the exact location.
- Intersection or Point & Click selection methods will provide more accurate results.

Select Your Destination By:

- ☐ Address
- ☐ Landmark
- ☐ Intersection
- ☐ Point & click

- Examples of Selection Methods

Select Origin & Destination

Reset

Rajah 2.8: Antaramuka Bus Information System

Merujuk kepada rajah diatas, input lokasi (*select your starting point*) dan destinasi (*select your destination*) boleh dibahagikan kepada 4 kategori iaitu :-

- Alamat (*Address*) – pengguna memasukkan alamat yang ingin dituju
- Mercutanda (*Landmark*) – pengguna memilih mercutanda daripada pilihan yang sedia ada. Cth: tempat bersejarah, balai polis, pusat beli-belah dan lain-lain lagi.
- Persimpangan (*Intersection*) – pengguna memilih persimpangan daripada pilihan yang sedia ada.

- Point & Click – pengguna memilih lokasi/destinasi atas peta maya.

2.7.1 *Kebaikan dan kelebihan Bus Information System*

- Dua versi: *Web-based* dan *stand-alone*.
- Lokasi dan destinasi berdasarkan kepada input pengguna.
- Pilihan lokasi dan destinasi yang sedia ada berdasarkan perhentian bas, *landmark*, *intersection* dan *point & click*.
- Antaramuka yang menarik dan mudah difahami.
- Hasil pengiraan ditunjukkan diatas peta maya dan secara tulisan.
- Anggaran masa tiba diberikan.
- Pengiraan laluan terpendek berbeza mengikut hari dan masa. Sebagai contoh, pada hujung minggu, bilangan dan kekerapan bas mungkin berkurang, begitu juga jika pengguna menggunakan perkhidmatan di waktu malam.
- Informasi mengenai peta kawasan serta laluan bas yang terperinci dan terkini.

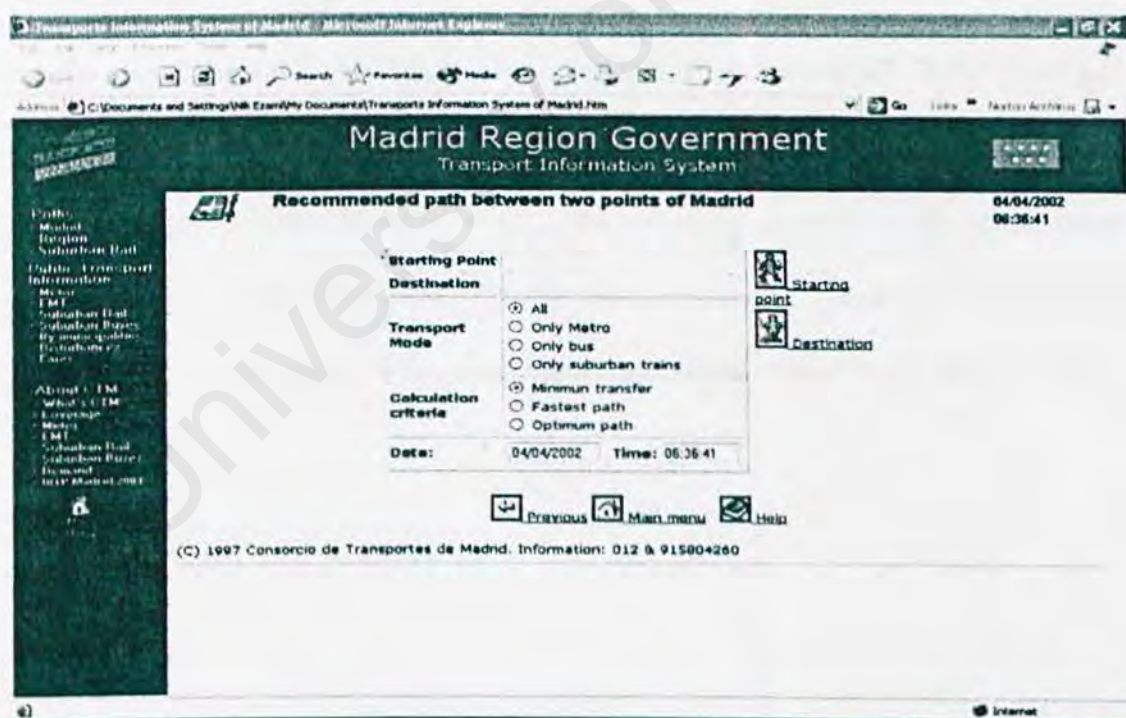
2.7.2 Kelemahan Bus Information System

Pilihan yang terhad

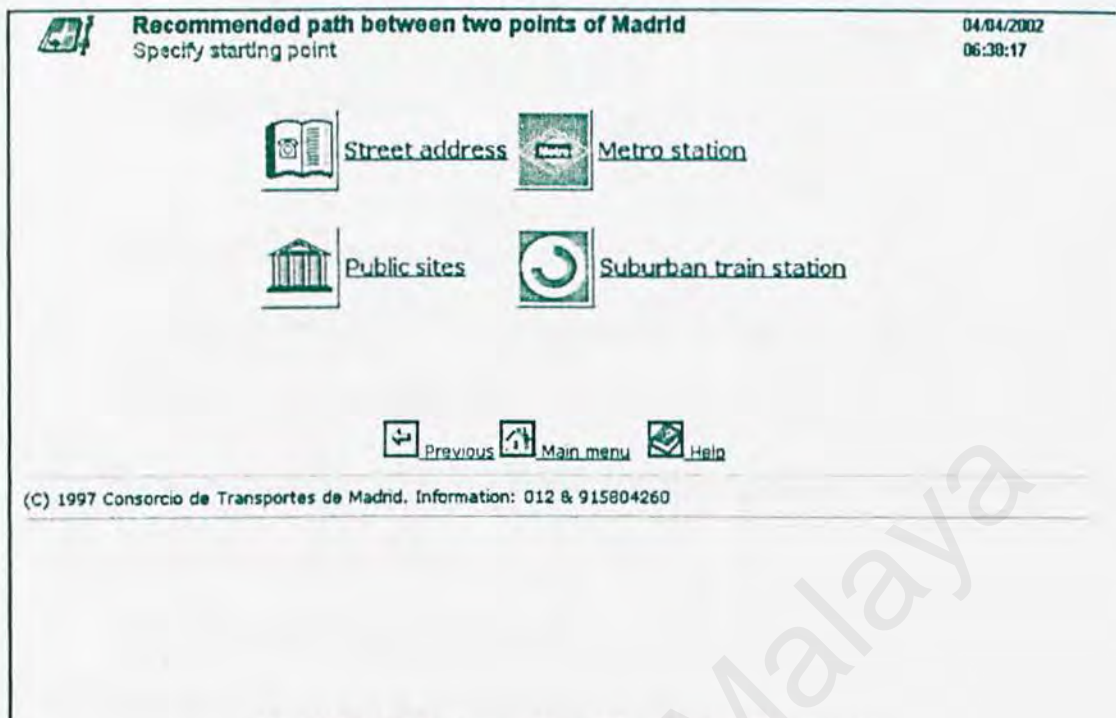
Bus Information System mempunyai pilihan pengguna yang agak terhad. Sebagai contoh, sistem ini hanya akan mengira laluan bas yang terpendek. Laluan bas yang terpendek ini mungkin memerlukan pertukaran bas yang banyak dan masa yang lama. Jadi, saya rasa bahawa sistem ini perlu menambahkan lagi pilihan supaya sistem dapat mengira laluan bas yang mengambil masa yang paling cepat dan laluan bas yang paling kurang pertukaran bas.

2.8 Madrid Region Government TRANSPORT INFORMATION SYSTEM

Perisian ini dibangunkan oleh EnWare untuk *Consortio de Transportes de Madrid* iaitu jabatan pengangkutan daerah Madrid. Laman web ini menyediakan pelbagai kemudahan untuk menyenangkan pengguna yang menggunakan mod pengangkutan yang berbeza. Laman web ini secara langsung mempunyai informasi mengenai semua jenis pengangkutan yang terdapat di daerah Madrid terutamanya bas. Bagi mencari laluan bas yang akan digunakan, pengguna haruslah menginputkan lokasi dan destinasi yang dituju.



Rajah 2.9: Antaramuka bagi *Madrid Region Government Transport Information System*.



Rajah 2.10: Antaramuka pilihan lokasi/destinasi

Lokasi dan destinasi boleh dimasukkan oleh pengguna atau boleh dipilih daripada pilihan yang diberi sepertimana yang ditunjukkan oleh rajah diatas. Setiap pilihan mempunyai sub-pilihan. Sebagai contoh, jika pengguna memilih *public sites*, antara sub-pilihan yang akan keluar ialah hospital, sekolah, kerajaan, penginapan dan lain-lain lagi. Setelah pengguna memilih sub-pilihan yang dikehendaki, suatu senarai pilihan akan dipaparkan.

2.8.1 Kebaikan dan kelebihan Madrid Region Government Transport Information System.

- Antaramuka yang menarik dan sistematik
- Pilihan pengguna (user's option) yang banyak dari segi lokasi dan destinasi
- *Web-based* – mudah dicapai
- 3 jenis algoritma pengiraan – laluan terpantas, pertukaran bas yang paling kurang, dan laluan optimum.
- Banyak informasi mengenai laluan bas
- Mempunyai Pengkalan data yang besar

2.8.2 Kelemahan Madrid Region Government Transport Information System.

Tidak mempunyai kemudahan point & click

Transport Information System ini tidak mempunyai peta maya yang membolehkan pengguna memilih lokasi dan destinasi mereka berdasarkan kepada peta maya kawasan mereka. Peta maya ini amat berguna terutamanya kepada pengguna bas yang baru.

Tidak memaparkan laluan bas pada peta

Sistem ini tidak memaparkan laluan bas pada peta maya. Laluan bas yang ditunjukkan diatas peta maya akan memudahkan pengguna. Pengguna dapat mengetahui dan memahami keseluruhan laluan bas. Kemudahan ini amat berguna jika digabungkan dengan kemudahan *point & click*.

BAB 3

METODOLOGI PEMBANGUNAN SISTEM

3.1 PENGENALAN

Disiplin kejuruteraan perisian penting dalam pengendalian dan pelaksanaan pembangunan sesuatu sistem. Kitar Hayat Pembangunan Sistem (*System Development Life Cycle*) adalah metodologi pembangunan sistem yang telah dipiawaikan bagi memastikan proses pembangunan mematuhi standard yang ditentukan. Kitar hayat (Kitar Hayat Pembangunan Sistem) ini dilihat sebagai hayat sesuatu produk perisian daripada peringkat konsep sehingga ke peringkat implementasi, penghantaran, penggunaan dan penyelenggaraan. Jujukan langkah-langkah pembangunan yang teratur dalam kitar hayat adalah bertujuan untuk memastikan proses pembangunan konsisten dan berkualiti.

3.2 METODOLOGI

Pembangunan sistem dengan menggunakan pendekatan Kitar Hayat Pembangunan Sistem adalah terbahagi kepada beberapa model proses perisian seperti model Air Terjun (*waterfall*), model V, model Prototaip.

Bagi proses pembangunan sistem ini, model Air Terjun (*Waterfall*) dengan Prototaip telah digunakan. Model ini secara asalnya telah wujud sejak 1970, tetapi model Air Terjun dahulu mempunyai beberapa masalah. Model Air Terjun yang kita ketahui sekarang adalah model Air Terjun yang telah diperbaharui.

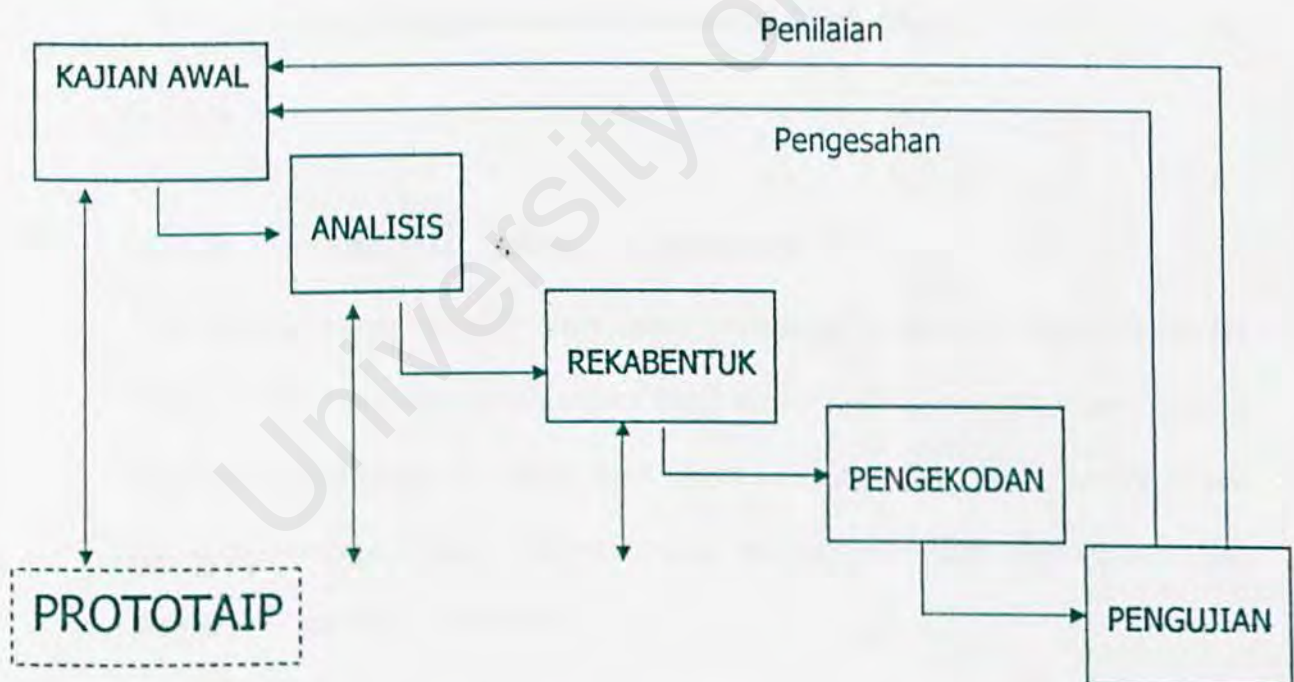
Dengan adanya prototaip, ia membolehkan pengguna serta pembangun untuk menguji/menganalisa sistem yang separa siap supaya mereka boleh memnentukan aspek sistem yang sesuai untuk diimplementasikan dalam sistem terakhir. Sebagai contoh, pembangun boleh membina suatu sistem untuk mengimplementasikan sebahagian kecil keperluan utama (*key requirement*) untuk memastikan keperluan tersebut konsisten dan praktikal; jika tidak, pengubahsuaian akan dilakukan pada fasa kajian awal, berbanding dengan pengubahsuaian kos tinggi pada fasa pengujian. Model ini mempunyai beberapa kelebihan bagi memastikan proses pembangunan sesebuah sistem itu teratur dan mengikut pawai serta hasil daripada projek ini berkualiti antaranya:

- Memberi pandangan peringkat tinggi terhadap proses yang sedang berlaku dalam pembangunan sistem. Memudahkan pemahaman terhadap proses pembangunan sistem terutamanya kepada pelanggan yang tidak biasa.
- Memberikan pembangun jangkaan jujukan aktiviti/fasa yang mereka akan hadapi kelak.
- Mudah untuk mengukur penggunaan masa bagi sesuatu aktiviti/peringkat itu.
- Proses prototaip digunakan untuk mengawal setiap peringkat dimana ia dapat membantu menafsir strategi-strategi rekabentuk yang lain.
- Penggunaan kaedah pengesahan (*validation*) bagi memastikan sistem mengimplementasikan semua keperluan, supaya setiap fungsi sistem boleh dijejak ke keperluan tertentu dalam spesifikasi.
- Penggunaan ujian pengesahan (*verification*) keperluan bagi memastikan setiap fungsi berjalan dengan betul.

Pembangunan sistem ini adalah dibuat berdasarkan kepada 5 fasa. Fasa-fasa tersebut ialah:

- *Fasa Kajian Awal*
- *Fasa Analisa Sistem*
- *Fasa Rekabentuk Sistem*
- *Fasa Pengkodan/Pembangunan*
- *Fasa Pengujian dan Perlaksanaan*

Rajah 3.1 - menunjukkan model Air Terjun dengan Prototaip.



Rajah 3.1: Fasa Pembangunan Projek

3.2.1 FASA I: KAJIAN AWAL

Fasa ini dikenali sebagai Kajian Kesauran (*feasibility study*). Fasa ini juga boleh diterangkan sebagai Fasa Penyiasatan dan Mengenalpasti.

Fasa Penyiasatan dan Mengenalpasti

Fasa ini merupakan kajian awal bagi mendapatkan maklumat dan gambaran tentang sistem yang akan dibangunkan. Fasa ini bertujuan untuk mendefinisikan objektif bagi Sistem Informasi direktori bas dalam menggantikan sistem manual yang sedia ada. Skop pembangunan sistem juga telah dikenalpasti supaya pembangunan dapat memenuhi objektif. Siasatan ke atas sistem manual telah dilakukan bagi mendapatkan masalah-masalah yang timbul melalui penggunaannya. Kehendak serta pendapat pengguna sistem juga di ambilkira pada fasa ini.

3.2.2 FASA II: ANALISA SISTEM

Fasa analisa sistem pula bertujuan untuk memahami bagaimana Sistem Informasi Direktori Bas yang akan dibangunkan dapat menyelesaikan masalah sistem manual yang telah dikenalpasti dari kajian awal, ianya meliputi kajian sistem, keperluan dan spesifikasi sistem, cadangan pengisian sistem, analisis alatan pembangunan sistem dan pengantaramuka yang digunakan.

3.2.2.1 Analisa Sistem

- Sistem yang dibangunkan adalah sistem atas-talian (*on-Line*), dimana pembangunan sistem berlaku dalam komputer sendiri dan akan dimuat-naik ke server dimana ia boleh diakses oleh pengguna: -
 - Capaian secara atas-talian (*on-line*) lebih cepat berbanding secara manual
 - Boleh dicapai oleh semua pengguna tidak kira tempat
 - Kapasiti maksimum data yang dapat dimuatkan tidak tertakluk kepada nilai tetap (boleh berubah)
 - Kos yang agak murah
 - Persembahan datanya yang terdiri daripada antaramuka yang menarik

3.2.2.2 Analisa Keperluan Sistem

Keperluan menggambarkan sifat/aktiviti sesuatu sistem. Analisa Keperluan boleh dibahagikan kepada dua iaitu :

- *Keperluan Berfungsi*
- *Keperluan Tak Berfungsi*

Keperluan Berfungsi

Keperluan berfungsi ialah fungsi atau kebolehan yang boleh dilakukan oleh sesuatu sistem itu. Keperluan berfungsi juga menerangkan tentang interaksi antara sistem dan persekitarannya dan bagaimana sistem akan bertindak terhadap rangsangan/stimuli tertentu.

Fungsi Lokasi

Lokasi perlu dimasukkan oleh pengguna sebagai titik permulaan pencarian. Lokasi ini ialah nama-nama stesyen/perhentian bas yang sedia ada dalam sistem. Pengguna hanya perlu memilih stesyen/perhentian yang ada melalui *drop-down menu*.

Fungsi Destinasi

Destinasi perlu dimasukkan oleh pengguna supaya pencarian lengkap. Destinasi juga hanya terdiri daripada stesyen/perhentian bas. Pengguna hanya perlu memilih stesyen/perhentian yang ada melalui *drop-down menu*.

Fungsi Point & Click

Fungsi ini membolehkan pengguna memilih lokasi dan destinasi berdasarkan peta interaktif. Pengguna perlu memilih lokasi diatas peta dan tekan padanya. Kemudian pengguna memilih destinasi yang diinginkan dan tekan padanya. Fungsi ini berkait secara langsung dengan fungsi lokasi dan fungsi destinasi dimana nod-nod (lokasi & destinasi) yang dipilih pada peta akan sama dengan fungsi input (lokasi & destinasi).

Fungsi Pengiraan Laluan

Fungsi ini akan mengira laluan terpantas berdasarkan lokasi dan destinasi yang diinputkan pengguna. Fungsi ini juga akan mengambil kira kemungkinan yang akan berlaku apabila mengikuti laluan terpantas iaitu pertukaran bas.

Fungsi Paparan

Fungsi ini akan memaparkan segala hasil dalam dua tetingkap yang berbeza: -

- *Tetingkap Peta Interaktif*

Tetingkap ini memuatkan peta interaktif kawasan yang dipilih. Tetingkap ini akan berada dibahagian atas skrin. Dalam tetingkap ini, pengguna boleh memasukkan input melalui fungsi *point & click* dan hasil pengiraan laluan antara dua nod akan dipaparkan dalam bentuk grafik.

- *Tetingkap Laluan*

Tetingkap laluan akan memaparkan hasil pengiraan laluan menggunakan perkataan. Hasil laluan ini mengandungi lokasi untuk mengambil bas,

nombor bas serta laluan, dimana perlu berhenti dan pertukaran bas yang perlu dibuat.

Fungsi Direktori

Fungsi ini dapat memaparkan maklumat-maklumat mengenai nombor serta laluan bas, jangkaan waktu, kekerapan, maklumat mengenai perhentian/stesyen bas, maklumat mengenai syarikat bas, tambang bas dan lain-lain lagi. Maklumat yang terkandung dalam fungsi direktori ini tidak berkaitan dengan pengiraan laluan terpantas. Maklumat direktori ini ditunjukkan secara keseluruhan.

Keperluan Tak Berfungsi

Keperluan tak berfungsi adalah kekangan dimana sistem mesti beroperasi untuk mengatasi kekangan ini. Dengan kata lain, keperluan tak berfungsi akan membatasi pilihan yang ada untuk menyelesaikan suatu masalah supaya ia berada dalam keadaan yang terkawal (mengurangkan aspek luar jangkaan) dan relevan dengan keperluan.

Keperluan tak berfungsi bagi sistem ini adalah seperti berikut:

Kebolehpercayaan

Sistem ini boleh dipercayai iaitu ia tidak akan menghasilkan kos yang tinggi apabila digunakan dengan cara yang betul dan mengikut prosedurnya.

Keselamatan

Mempunyai ciri-ciri keselamatan ketika menerima masukan data atau capaian dibuat seperti kawalan katalaluan.

Keringkasan

Skrin dan arahan diatur dengan tersusun bagi memudahkan pemahaman pengguna dan boleh menarik minat pengguna

Keberkesanan

Keberkesanan bermaksud skrin input dan output mempunyai tujuan yang khusus dalam sistem.

Antaramuka yang menarik

Daya penarik dalam sesuatu sistem adalah penting kerana ia dapat menarik minat pengguna untuk menggunakan sesuatu sistem itu.

3.2.2.3 Analisa Keperluan Perisian

Pemilihan perisian yang bersesuaian untuk pembangunan sistem adalah penting bagi mengimplementasikan sistem. Ini bertujuan untuk memastikan bahawa perisian yang digunakan adalah bersesuaian dengan konsep projek ini. Pemilihan ini termasuklah perisian utama untuk pembangunan sistem dan perisian-perisian lain yang digunakan untuk memberikan ciri-ciri tambahan seperti perisian untuk pemprosesan kod bar sebagainya. Perisian yang digunakan untuk membangunkan sistem rekod penyewaan video ini adalah:

- *Microsoft® Access™ 2002*
- *Microsoft® Visual Basic™ 6.0*
- *Microsoft® Visual Interdev™ 6.0*
- *Adobe® Photoshop™ 7.0*

Microsoft Access 2002

Setelah menjalan pemerhatian ke atas perisian-perisian yang sesuai digunakan untuk pembangunan sistem, perisian Microsoft Access 2002 dipilih untuk pembangunan pangkalan data bagi Sistem Informasi Direktori Bas. Ia sesuai digunakan oleh pengaturcara pangkalan data yang berpengalaman malah oleh pengguna pangkalan data yang baru. Microsoft Access 2002 memperkenalkan pelbagai alatan untuk membangunkan, mentadbir serta menganalisa data. Dengan sokongan XML, pengguna boleh mencipta pangkalan data yang kompleks yang mana ia boleh diintegrasikan dengan senang kedalam web. Pengguna baru akan menemui alatan yang serba lengkap supaya pembangunan pangkalan data menjadi lebih mudah dan menjimat masa. Ianya mudah dipelajari dan senang dilaksanakan. Pengenalan ciri baru seperti Microsoft PivotTable® dan Microsoft PivotChart® meningkatkan lagi kebolehan menganalisa data pengguna.

Kesimpulannya, Perisian ini akan diguna khas untuk mencipta dan mentadbir pangkalan data sistem informasi direktori bas.

Visual Basic 6.0

Visual Basic 6.0 adalah perisian pengaturcaraan berorientasikan objek. Perisian ini mempunyai 3 komponen yang utama iaitu *object*, *properties*, dan *method*.

- *Object* adalah benda atau nama
- *Properties* menerangkan sifat-sifat objek seperti nama, saiz dan warna
- *Methods* ialah fungsi yang dilakukan oleh objek seperti *move*, *print*, *resize*, *calculate* dan *clear*.

Visual Basic juga menggabungkan satu set teknologi perisian yang dipanggil Active X. teknologi ini membenarkan ciptaan, integrasi, penggunaan semula komponen perisian yang dipanggil kawalan. Ciri-ciri yang terdapat pada Visual Basic 6.0 :

- Mempunyai ciri-ciri antaramuka pengguna yang menarik
- Dapat diintegrasikan atau diautomasikan dengan aplikasi lain seperti ODBC, Excel, FoxPro, Paradox, dan terutamanya dengan Access.
- Dapat merekabentuk graf dan carta secara program serta memproses graf atau jenis data yang lain.
- Visual Basic dapat diperkembangkan dengan menambah *Custom Control* dan memanggil prosedur dalam *Dynamic Link Library* (DLL).

Visual Basic juga membolehkan pembangunan Web dengan menggunakan teknologi *Microsoft IntelliSense®* dan pelengkap tag (tag HTML), atau memilih editor

WYSIWYG (what you see is what you get) untuk kemudahan drag-and-drop dalam pembinaan aplikasi Web.

Kesimpulanya, Perisian ini akan digunakan tertamanya bagi membentuk pelbagai komponen serta fungsi yang akan digunakan dalam sistem mengikut keperluan yang telah dicatatkan.

Microsoft Visual InterDev 6.0

Microsoft® Visual InterDev™ ialah alatan pembangunan web yang telah direka untuk pengaturcara yang ingin mencipta:

- Aplikasi Web yang berorientasikan data menggunakan sumber data yang disokong oleh ODBC atau OLE DB, seperti DMS (database management system) dari Microsoft.
- Laman Web capaian luas menggunakan HTML dan *script* dalam aplikasi web yang menggunakan sepenuhnya perkembangan terkini dalam teknologi pelayar (*browser*), seperti Microsoft® Internet Explorer 4.0, Dynamic HTML dan ciri-ciri multimedia. Menjamin pengoperasian sepenuhnya dengan sebarang pelayar.
- Persekitaran Pembangunan yang tahan lasak dengan Scripting Object Model, design-time controls (DTCs), dan toolbox mudah-alih untuk pembangunan yang pantas, pengujian dan pengawasilap untuk membuang pepijat pada laman. Ini

menggalakkan pembangunan aplikasi web dengan pantas dan memudahkan pengaturcaraan.

- Integrasi Penyelesaian yang boleh terdiri daripada applet dan komponen yang dicipta Microsoft[®] Visual Basic[®], Visual C++[®], Visual J++[™], dan Visual FoxPro[®]. InterDev dapat mengintegrasikan komponen yang telah dicipta dengan Visual Basic dengan senang. Komponen tersebut tidak perlu melalui sebarang konversi.

Microsoft InterDev 6.0 akan digunakan bersama komponen Visual Basic dan pangkalan data yang dicipta dengan Access untuk membentuk aplikasi web.

Adobe Photoshop 7.0

Perisian ini digunakan khas untuk mencipta, memperbaiki dan mengubahsuai pelbagai format grafik/imej. Dalam pembangunan sistem ini, perisian ini akan digunakan dalam pelbagai aspek dalam pembangunan peta interaktif, antaramuka sistem informasi. Adobe Photoshop menyediakan alatan yang lengkap bagi memenuhi semua kehendak pengguna/artis berpengalaman mahupun yang baru. Ia membolehkan saya untuk melakukan kerja dengan lebih efisien, menerokai idea-idea kreatif dan mencipta grafik yang berkualiti tinggi untuk kegunaan web. Antara kemudahan dan kelebihan yang disediakan:

- Lebih kurang 95 penapis kesan khas
- Siarkan imej *zoom* dan *pan* yang beresolusi tinggi bagi sebarang laman web dengan mudah melalui zoompoint zoomview.

3.2.2.4 Analisa Keperluan Perkakasan

Perkakasan biasanya merujuk kepada mesin atau peralatan fizikal yang melakukan fungsi-fungsi asas yang terkandung di dalam kitaran pengoperasian proses. Spesifikasi minimum yang diperlukan untuk melarikan Sistem Informasi Direktori Bas adalah seperti berikut:

- Pemprosesan Intel Pentium 166 atau lebih
- 32+MB RAM memori
- 300 MB ruangan cakera keras minimum
- Pelayar Internet Explorer/Netscape Navigator
- Tetikus
- Windows 98/2000/XP

3.2.2.5 Analisis Rekabentuk Antaramuka

- Paparan dan arahan hendaklah jelas, dengan menyediakan menu yang dilabel baik dengan menggunakan grafik, ikon dan teks yang mudah difahami.
- Antaramuka pengguna hendaklah mementingkan dan memudahkan interaksi yang berkesan
- Penampilan yang konsisten dan tetap (*standard*).
- Maklum balas dari sistem mestilah baik dan menggalakkan interaksi dengan pengguna.

3.2.2.6 Analisis Keupayaan Sistem

- Kebolehpercayaan terhadap sistem dicapai oleh pengguna tanpa ralat.
- Keupayaan memaparkan kandungan data.
- Keupayaan menganalisis input pengguna.
- Keupayaan melakukan perancangan laluan dengan tepat dan sistematik.

3.2.3 FASA III: REKABENTUK SISTEM

Rekabentuk adalah proses kreatif yang memerlukan pemahaman dan kebolehan semulajadi perekabentuk bagi menukarkan masalah kepada sesuatu bentuk penyelesaian. Ia mestilah dilatih dan dipelajari berdasarkan pengalaman dan mengkaji sistem yang sedia ada.

Fasa ini melibatkan penerangan tentang keseluruhan sistem yang dibangunkan dan jangkaan pencapaian sistem. Dalam fasa ini semua ciri-ciri sistem seperti senibina sistem, rekabentuk pangkalan data, rekabentuk proses seperti carta struktur dan diagram aliran data serta rekabentuk antaramuka pengguna dihuraikan, antaranya rekabentuk menu, persembahan kandungan, modul, kamus data dan pangkalan data. Kesemua ini memudahkan lagi penyediaan rekabentuk paparan untuk sistem yang dibangunkan. Fasa ini dihuraikan selengkapnya dalm bab 4.

3.2.4 FASA IV : PENGATURCARAAN / PENGKODAN

Pengaturcaraan/Pengkodan merupakan proses yang terpenting dalam membangunkan sesuatu sistem. Ia akan menentukan samada sesuatu sistem itu berjaya mencapai objektifnya atau tidak. Tanpa pengaturcaraan, sistem tidak akan lengkap sepenuhnya. Oleh itu, persekitaran pembangunannya adalah penting bagi menjamin proses pengaturcaraan ini dapat dilakukan dengan sempurna. Segala kajian dan analisis telah dilakukan dengan teliti bagi menentukan persekitaran pembangunan yang paling sesuai untuk projek ini. Selain itu terdapat 3 aspek utama yang dititikberatkan didalam proses pengkodan ini iaitu struktur kawalan, algoritma (kelas algoritma yang akan digunakan untuk pengkodan komponen yang digunakan) dan struktur data (proses format dan simpanan data supaya mentadbir dan manipulasi data menjadi mudah). Perlaksanaan fasa ini akan bermula pada semester akan datang.

3.2.5 FASA V: PENGUJIAN DAN PENYELENGGARAAN

Proses Pengujian ini dilaksanakan untuk menguji keberkesanan sesuatu aturcara itu menjalankan fungsinya. Ia bertujuan untuk mencari ralat pada sesuatu sistem itu dan menjejaki kesilapan aturcara. Dengan ini, ia dapat memastikan modul-modul serta hubungannya yang dibina adalah bebas daripada sebarang masalah supaya sistem akan dapat memberikan keputusan yang baik dan berkesan. Dalam fasa ini juga, sebarang algoritma, rekabentuk yang telah dilakukan sepanjang proses pembangunan sistem dapat dibuat penelitian dan penilaian semula jika tidak menepati keperluan yang ditetapkan.

Proses pengujian juga merupakan fasa yang paling penting bagi memastikan samada sistem yang dihasilkan telah memenuhi kehendak pengguna atau tidak. Sistem yang telah diimplementasikan dengan betul mampu menjalani apa jua pengujian yang diberikan.

Penilaian akan memastikan bahawa sistem telah mengimplementasikan kesemua keperluan, supaya setiap fungsi sistem boleh dirujuk kepada keperluan tertentu dalam spesifikasi.

Pengesahan akan memastikan bahawa setiap fungsi berfungsi dengan betul.

Fasa 5 ini akan dilaksanakan selepas fasa implementasi/pengaturcaraan.

BAB 4

REKABENTUK SISTEM

4.1 PENGENALAN

Pengguna atau pelanggan biasanya memerlukan suatu sistem baru disebabkan tiada sistem yang sedia ada atau ada aspek-aspek yang perlu diperlu ada dalam sistem lama tetapi tidak diimplementasikan. Rekabentuk merupakan satu proses kreatif yang menukarkan masalah kepada penyelesaian. Ia digambarkan sebagai satu proses yang menjadi perwakilan kepada struktur data, struktur program, ciri-ciri antaramuka dan maklumat-maklumat berprosedur. Rekabentuk merupakan satu kaedah utama yang dapat menafsirkan keperluan pengguna kepada produk perisian atau sistem akhir. Segala maklumat yang dikumpulkan semasa fasa analisa sistem akan diubah dalam bentuk modul-modul yang kecil sehinggalah terbentuknya sebuah sistem.

4.2 REKABENTUK STRUKTUR

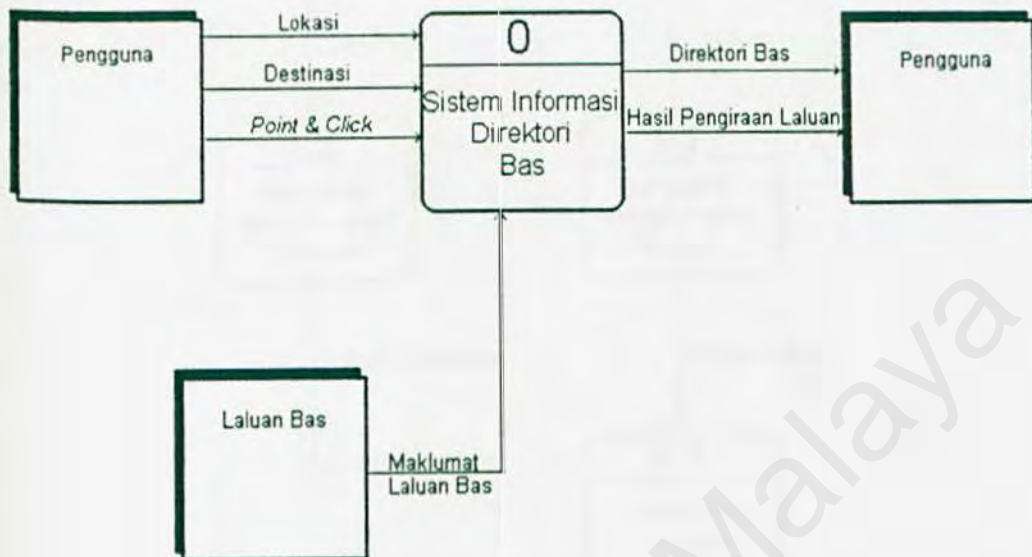
Untuk menggambarkan struktur Sistem Informasi Direktori Bas, carta struktur akan digunakan. Merujuk kepada carta yang berikut, Sistem Informasi Direktori Bas ini telah dibahagikan kepada komponen-komponen mengikut fungsian. Komponen yang berada di paras atas (form Selamat Datang) akan mewakili fungsi-fungsi yang akan ditemui di paparan awal antaramuka sistem dan diikuti dengan komponen-komponen seterusnya.



Rajah 4.1: Carta Struktur Sistem Informasi Direktori Bas

4.3 REKABENTUK PROSES

Rekabentuk proses pula akan digambarkan menggunakan Rajah Aliran Data (DFD). Ia merupakan satu teknik bergrafik yang dapat memaparkan perjalanan data-data dalam sistem. Selain itu, ia dapat memaparkan proses perubahan atau penukaran yang diimplementasi ke dalam data apabila data masuk dan keluar daripada sistem. Ia boleh dibahagikan kepada paras-paras yang dapat menggambarkan peningkatan perjalanan data dan maklumat-maklumat fungsian.



Rajah 4.2: Gambarajah Konteks



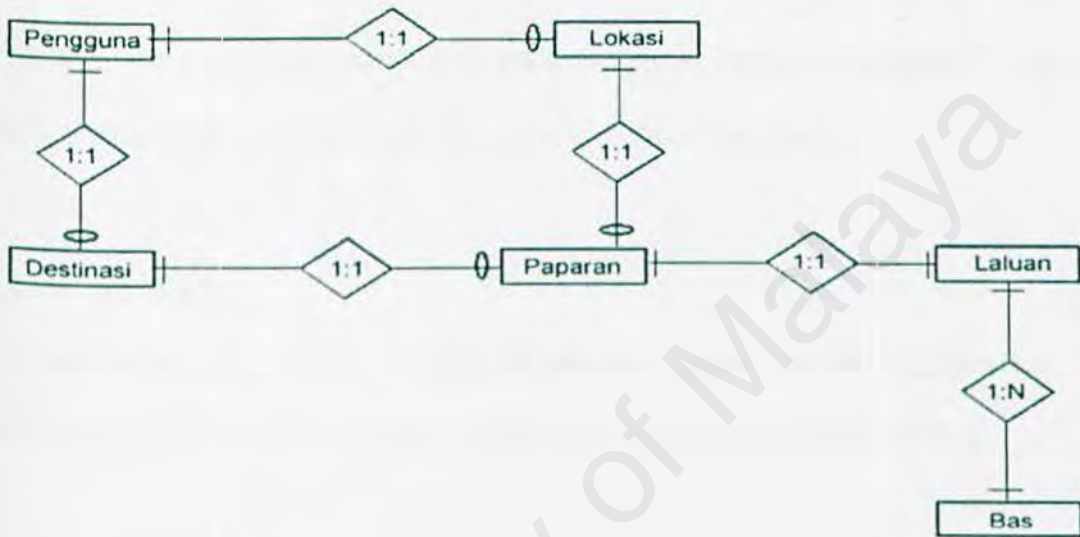
Rajah 4.3: Aliran Data Proses Perancangan Laluan

4.4 REKABENTUK PANGKALAN DATA

Dalam merekabentuk pangkalan data bagi Sistem Informasi Direktoi Bas, terdapat beberapa entiti yang telah dikenalpasti dan perlu direka.

4.4.1 Rajah Hubungan Entiti (ER)

Rajah Hubungan Entiti akan digunakan bagi menunjukkan setiap jenis hubungan di antara entiti-entiti.



Rajah 4.4: Hubungan Entiti (ER)

Berikut pula adalah senarai unsur (“element”) XML bagi menunjukkan hubungan yang telah dibina menggunakan kunci-kunci:

- Unsur NODE
- Unsur LINKS

4.4.2 Jadual Pangkalan Data

Pangkalan data bagi Sistem Informasi Direktori Bas akan dibangunkan menggunakan bahasa pengaturcaraan XML. Terdapat beberapa “*element*” yang digunakan dalam penyelarasan pangkalan data. Semua data-data telah disimpan ke dalam kumpulan “*element*” yang akan menggambarkan entiti-entiti yang berbeza. Kumpulan-Kumpulan berikut akan memberi penerangan tentang atribut yang digunakan:

Unsur bagi NODE

“Node” mewakili unsur-unsur yang menerangkan perhentian bas, stesyen bas dan persimpangan. Dibawah ditunjukkan contoh sub unsur dalaman dalam NODE:

- Unsur pertama

<X>6000</X>

Menerangkan kedudukan node pada (mewakili perhentian bas, stesyen bas dan persimpangan) paksi-X.

- Unsur kedua

<Y>300</Y>

Menerangkan kedudukan node pada (mewakili perhentian bas, stesyen bas dan persimpangan) paksi-Y.

- Unsur ketiga

<ID>1</ID>

Unsur ini memberi nilai ID (integer) pada node tersebut.

- Unsur keempat

<Nama>Jln_Tun_Mohd_Fuad</Nama>

Unsur ini memberi nama pada node tersebut.

- Unsur kelima

<jenis>p</jenis>

Menerangkan jenis node tersebut samaada ianya adalah persimpangan atau perhentian/stesyen bas. Persimpangan diwakili dengan aksara “j” manakala perhentian diwakili dengan aksara “p”.

Kesemua sub unsur ini akan dimuatkan dalam dalam unsur NODE. Contoh unsur NODE dalam dokumen XML-jalah:

```
<Node><X>6000</X><Y>300</Y><ID>1</ID><Name>Jln_Tun_Mohd_Fuad</Name><jenis>p</jenis></Node>
```

Rajah 4.5: Contoh unsur NODE dalam XML

Unsur bagi LINKS

Unsur LINKS pula menerangkan sub unsur yang berkaitan dengan hubungkait antara unsur NODE. Unsur LINKS ini secara amnya menerangkan sambungan diantara NODE iaitu sambungan diantara perhentian bas atau persimpangan. Sub unsurnya ialah :-

- Unsur pertama

<Nd1>1</Nd1>

ID node pertama dalam siri hubungan. Hubungan adalah antara dua node sahaja dalam satu unsur LINKS.

- Unsur kedua

<Nd2>2</Nd2>

ID node kedua dalam siri hubungan. Ia melengkapkan hubungan.

- Unsur ketiga

<Kos>10</Kos>

Kos bagi sambungan/hubungan tersebut. Ini digunakan sebagai jarak maya dalam pengiraan laluan terpantas.

- Unsur kelima

<ID>1-2</ID>

ID bagi unsur LINKS yang dengan jelasnya menyatakan bahawa sambungan/hubungan adalah diantara NODE (ID : 1) dengan NODE (ID : 2).

Keseluruhan unsur LINKS ini adalah seperti rajah dibawah :

```
<Link><Nd1>1</Nd1><Nd2>2</Nd2><Cost>10</Cost><ID>1-2</ID></Link>
```

Rajah 4.6 : Contoh unsur LINKS dalam XML.

4.5 REKABENTUK ANTARAMUKA

Rekabentuk antaramuka pengguna adalah amat penting dalam pembangunan sesebuah perisian. Ini kerana rekabentuk yang menarik akan menambahkan lagi minat pengguna untuk mencuba sesuatu perisian itu.

Objektif antaramuka pengguna adalah :

- Kecekapan antaramuka

Kecekapan antaramuka dalam mengendalikan kelajuan penyimpanan data dan ada kesalahan

- Pertimbangan pengguna

Antaramuka yang dihasilkan mestilah memenuhi keperluan pengguna dan respon yang bertepatan mesti diberikan oleh sistem kepada pengguna.

- Produktiviti

Ini dapat diukur melalui penghasilan antaramuka yang baik yang dapat menjimatkan kos dan masa pengguna.

- Keberkesanan penggunaan antaramuka

Ini dapat dicapai melalui rekabentuk antaramuka yang membenarkan pengguna menggunakan sistem mengikut keperluan mereka tanpa ada kekeliruan

Rekabentuk antaramuka dapat menggambarkan lakaran paparan yang akan digunakan sebagai medium interaksi antara pengguna dan mesin. Dalam merekabentuk antaramuka pengguna beberapa faktor perlu diambil kira, antaranya adalah :

- Menggunakan warna latarbelakang, ikon-ikon, logo, gambar dan huruf-huruf yang sesuai.
- Melakarkan halaman secara konsisten supaya pengguna tidak akan terkeliru.
- Menggunakan ciri-ciri kawalan seperti combo box, selection box, check box bagi memudahkan dan mempercepatkan proses masukkan dari pengguna.
- Menggunakan jenis huruf yang sesuai dan tidak terlalu bergaya bagi mengelakkan laman kelihatan tidak teratur dan sukar difahami.



The image shows a web form interface. It features a large, empty rectangular box at the top, likely for displaying search results or a map. Below this box is a horizontal bar containing four input fields: 'Lokasi' (Location) with a dropdown arrow, 'Destinasi' (Destination) with a dropdown arrow, 'Cari Lokasi' (Search Location) with a text input field, and 'Direktori' (Directory) with a text input field.

Rajah 4.7: Contoh Antaramuka Form Pilihan & Paparan

BAB 5

IMPLEMENTASI SISTEM

5.1 PENDAHULUAN

Implementasi suatu sistem ialah pembangunan sistem baru dengan berlandaskan keperluan serta objektif yang telah ditetapkan. Ia juga melibatkan penghantaran sistem tersebut dan seterusnya ke arah operasi. Operasi disini bermaksud penggunaan secara harian; working version. Implementasi sistem ialah kenyataan rekabentuk teknikal yang terkandung dalam rekabentuk sistem. Sistem maklumat membangunkan implementasi sistem yang merangkumi data, proses dan antaramuka pengguna berasaskan kepada perspektif pembangun sistem.

Implementasi sistem boleh dibahagikan kepada empat fasa utama iaitu:-

- i) Membina dan Menguji Pangkalan Data
- ii) Membina dan Menguji Program
- iii) Menginstall dan Menguji Sistem Baru
- iv) Menghantar Sistem Baru untuk Pengoperasian

5.1.1 Membina dan Menguji Pangkalan Data

Sekiranya aplikasi baru meminta rangkaian atau pangkalan data yang baru, ia biasanya akan diimplementasikan terlebih dahulu sebelum pembangunan atau penginstallan program komputer tersebut dilakukan. Pangkalan data yang digunakan untuk Sistem Maklumat Direktori Bas ialah dalam bahasa pengaturcaraan XML. Pangkalan ini diuji dengan teliti supaya setiap input atau output yang dikeluarkan adalah seperti yang dikehendaki.

Struktur pangkalan data diimplementasikan dan data telah sedia ada dalam pangkalan data. Pembangun program menulis dalam dokumen XML (memasukkan data) untuk mempopulasikan dan memantapkan pangkalan data. Pangkalan data diuji dengan teliti supaya setiap input atau output yang dikeluarkan adalah tepat.

5.1.2 Membina dan Menguji Program

Fasa ini juga dikenali sebagai fasa pembangunan. Fasa Pembinaan dan pengujian program ini biasanya memakan masa yang paling lama dan ia adalah fasa yang amat meletihkan di dalam pembangunan sistem. Pembangun program mestilah bekerja mengikut spesifikasi yang telah dibangunkan dan menapis menerusi fasa dan aktiviti yang sebelumnya dalam Model Air Terjun. Sekiranya spesifikasi adalah tidak jelas, tidak lengkap, tidak tepat atau sebaliknya rosak, fasa pembinaan akan menjadi lebih kompleks dan memakan masa yang lama.

Input kepada fasa ini ialah subset daripada kenyataan rekabentuk teknikal yang mana mengandungi spesifikasi program. Sekiranya, pangkalan data baru ataupun yang telah diubahsuai akan digunakan, pangkalan data yang tidak dipopulasi yang telah diimplementasikan adalah input daripada fasa implementasian yang sebelumnya. Produk daripada fasa ini ialah program komputer yang belum diinstall, yang mana program tersebut telah disemak (*debug*) dan diuji dengan sempurna tetapi ia belum lagi diinstall untuk penggunaan produksi.

5.1.3 Menginstall dan Menguji Sistem Baru

Dalam fasa ini proses menginstall dan menguji sistem baru dilakukan. Input utama kepada fasa ini ialah subset daripada kenyataan rekabentuk teknikal yang menspesifikasikan bagaimana program yang telah dibina dan diuji, fail dan juga pangkalan data akan dihimpun di dalam sistem yang diintegrasikan. Produk daripada fasa ini ialah sistem yang telah diinstall dan sedia untuk dihantar kepada produksi.

5.1.4 Menghantar Sistem Baru untuk Pengoperasian

Fasa terakhir implementasian ialah untuk menghantar sistem baru bagi pengoperasian. Pada kebiasaannya, sistem baru mewakili suatu peralihan daripada cara semasa sesuatu bisnis itu dijalankan. Oleh yang demikian, suatu perubahan yang lancar daripada sistem lama kepada sistem baru haruslah dilakukan bersama dengan bantuan tambahan kepada pengguna untuk menangani masalah *start-up* yang normal. Melatih pengguna dan menulis pelbagai manual pengguna dapat membantu pengguna dalam menggunakan sistem baru.

5.2 PEMBINAAN SISTEM INFORMASI DIREKTORI BAS

Fasa yang memerlukan masa yang paling lama dalam pembinaan SIDB ialah fasa pembinaan. Ia melibatkan penginterpretasian dan pengimplementasian keperluan-keperluan yang terkumpul termasuk rekaan teknikal sistem ke dalam kod-kod program. Pengabungan rekaan fizikal dan teknikal sistem ke dalam kod program telah dilakukan dengan menggunakan perisian Visual Basic 6.0.

5.2.1 Penyediaan Kandungan

Perkara yang paling penting pembinaan Sistem Informasi Direktori Bas ialah kandungan sistem iaitu apa yang akan dan perlu dipaparkan oleh sistem serta operasi yang akan diwujudkan untuk kegunaan pengguna. Sebahagian besar daripada kandungan sistem bergantung kepada keperluan sistem. Ia harus mengandungi kesemua ciri-ciri dan fungsi yang diperlukan oleh pengguna secara umum.

Dalam persekitaran pembangunan VB (EDI), Satu borang (*form*) yang utama telah digunakan. Borang ini menjadi asas kepada panggilan sub-sub borang yang lain lain. Sub-sub borang lain akan dipanggil melalui borang ini dan juga melalui modul (*module*). Modul digunakan khas untuk memaparkan borang pengenalan seperti *splash form* dan borang katalauan pengguna. Setiap sub borang mempunyai sub-modul/sub-fungsi yang unik.

Satu ciri yang terdapat dalam borang utama ini ialah penggunaan aplikasi *toolbar* yang dan *menu editor* sebagai medium untuk membuka sub-sub modul yag lain. *Icon*, istilah dan kunci jalan pintas (*short cut key*) yang mudah difahami digunakan untuk memudahkan pengguna sistem.

Untuk menjadikan sistem ini lebih menarik, warna, imej, latarbelakang berserta penggunaan butang yang bersesuaian dengan keperluan telah digunakan. Butang-butang ini boleh diperolehi daripada perisian Visual Basic 6.0.

5.2.2 Pengintegrasian Kandungan, Persembahan dan Pemrograman

Integrasi kesemua kandungan sistem yang terdiri daripada persembahan (GUI) dan pengaturcaraan adalah dengan menggunakan aplikasi Visual Basic 6.0. Proses ini dibahagikan kepada dua komponen utama iaitu :-

a) Komponen Visual (GUI)

b) Komponen Kod

Komponen antaramuka pengguna merujuk kepada skrin yang akan dipaparkan pengguna dan boleh berinteraksi dengan pengguna. Susunan serta rekabentuk setiap komponen utama ditentukan sendiri oleh pembangun sistem dengan mudah. Komponen ini direka dengan menggunakan borang (*form*) dan kawalan (*control*). Borang (*form*) dan kawalan (*control*) ini juga membenarkan pengendalian input pengguna, paparan maklumat, pilihan output dan keputusan pengguna. Dengan ini, persembahan dan pemrograman sistem boleh diintegrasikan bersama untuk menjanakan produk akhir.

Komponen pengkodan pula merujuk kepada baris-baris kod yang perlu ditaip oleh pembangun sistem dan ianya tidak dapat dilihat atau ditukar oleh pengguna.

5.2.3 Fasa Pengkodan

Fasa ini adalah fasa di mana semua hasil daripada fasa sebelumnya, iaitu fasa analisa dan rekabentuk direalisasikan kepada satu bentuk sistem aplikasi yang sebenarnya. Fasa ini memakan masa yang lama untuk disiapkan kerana SIDB ini dibangunkan dengan menggunakan bahasa pengaturcaraan peringkat tinggi. Untuk proses pembangunan SIDB, perisian pembangunan yang telah digunakan ialah Microsoft Visual Basic 6.0. Strategi serta pertimbangan yang sewajarnya perlu diaplikasikan terhadap kesemua semua output/hasil daripada fasa sebelumnya oleh pembangun sebelum ianya ditukarkan kepada sistem.

Dengan melakukan proses-proses tersebut dalam peringkat pengaturcaraan, ia telah menghasilkan modul-modul aturcara yang dikompilasikan dengan “bersih” iaitu tiada ralat. Ujian-ujian dilaksanakan terhadap setiap modul aturcara yang telah dikod dan dikompilasikan, agar setiap modul aturcara tidak mengandungi sebarang ralat.

Faktor-faktor kekangan terhadap pembangunan sistem juga perlu dipertimbangkan.

Antara faktor-faktor kekangan yang dipertimbangkan adalah :

- Faktor masa dan tenaga yang agak terhad bagi fasa ini
- Faktor sumber pembangunan yang terhad
- Faktor kos pembangun

- Faktor saiz sistem aplikasi ini yang perlu mempertimbangkan kemampuan komputer peribadi pengguna untuk melarikannya. (saiz aplikasi yang terlalu besar mungkin tidak dapat dilarikan dalam komputer peribadi pengguna)

5.2.4 Kaedah Pengkodan

Subsistem-subsistem yang terdapat di dalam sistem ini dibentuk berasaskan jujukan-jujukan fungsi, keperluan-keperluan data dan persamaan-persamaan logik. Setiap subsistem ini biasanya mempunyai satu atau beberapa aturcara yang berbeza.

Di dalam bahagian konsep rekabentuk aturcara berstruktur, konsep 'gandingan' akan menghasilkan aturcara-aturcara yang bermodul, sementara konsep 'ikatan' pula akan menghasilkan aturcara-aturcara yang berstruktur.

- **Pengaturcaraan Bermodul**

Pengaturcaraan bermodul merupakan suatu kaedah pengaturcaraan yang membahagikan satu masalah kompleks kepada bahagian-bahagian yang kecil agar mudah diaturcarakan. SIBD diaturcarakan seperti ini bagi mengatasi kekompleksan dan agar ianya mudah difahami.

- **Pengaturcaraan Berstruktur**

Pengaturcaraan berstruktur pula merupakan suatu kaedah pengaturcaraan yang teratur dan tertib. Di antara langkah-langkah pengkodan yang digunakan agar kaedah pengaturcaraan berstruktur dipatuhi ialah:

- i. Arah-an arahan cabang-an tanpa syarat perlu dihapuskan atau sekurang-kurangnya diminimumkan penggunaannya dalam setiap modul aturcara.
- ii. Arah-an arahan yang terkandung di dalam setiap rutin aturcara perlulah berasaskan suatu jujukan logik agar ia akan mengandungi hanya satu punca kemasukan dalam rutin dan punca keluar dari rutin.
- iii. Setiap rutin mestilah mengandungi kod-kod yang lengkap dan ditambah dengan komen-komen yang mudah difahami.

5.2.5 Pendekatan Pengkodan

Pendekatan pengkodan yang dipilih merupakan satu proses yang merumitkan dan agak kompleks kerana ia memerlukan komitmen yang tinggi dan usaha yang berterusan. Setiap modul (*form*) yang dihasilkan telah menggunakan pendekatan Model Air Terjun dengan prototaip manakala kawalan aliran projek menggunakan pendekatan bawah-atas (*bottom-up*). Kandungan dokumen sistem seperti laporan yang dipaparkan merupakan implementasi ketika fasa rekabentuk dan bukannya laporan yang dijana ketika masa larian sistem.

Dalam fasa ini juga, keupayaan pembangun teruji, yang mana setiap peristiwa (*event*) yang dijangkakan ke atas sistem memerlukan pembangunan kod secara manual (melalui pengetahuan pembangun).

5.3 RUMUSAN

Secara amnya, bab 5 ini menerangkan bahawa fasa pengkodan adalah fasa yang amat rumit dan kompleks di dalam proses pembangunan sesuatu sistem. Daripada proses pengkodan ini maka terhasillah antaramuka pengguna serta algotirma yang menjadi nadi sistem. Antaramuka (GUI) yang ramah pengguna membolehkan pengguna sistem dapat menggunakan sistem ini dengan lebih sistematik dan senang.

BAB 6

PENGUJIAN SISTEM

6.1 PENDAHULUAN

Bagi memastikan sesuatu perisian atau sistem itu berkualiti, pengujian sistem perlu dilakukan. Proses ini melibatkan penyemakan semula spesifikasi, rekabentuk dan pengkodan yang telah dijalankan sepanjang membangunkan sistem. Ianya merupakan satu proses yang kritikal dan harus dilaksanakan.

Antara tujuan pengujian dibuat juga adalah untuk memastikan bahawa modul-modul yang dibina bebas daripada sebarang ralat. Ralat yang wujud boleh menyebabkan ketidakbolehpercayaan pada sistem. Ralat juga boleh menyebabkan sistem kita tidak berfungsi sebagaimana yang dikehendaki spesifikasi iaitu melakukan apa yang sepatutnya dilakukan serta menghasilkan apa yang sepatutnya. Secara amnya, pengujian akan dilakukan dengan menggunakan data/input percubaan daripada pengguna.

Sesuatu ujian dikatakan baik jika ia berupaya untuk mengenalpasti ralat-ralat yang tidak dapat dikesan semasa fasa analisis, rekabentuk dan pengkodan. Objektif utama untuk pengujian sistem adalah:

- **Mengenalpasti ralat**

Pemeriksaan secara teliti dilakukan ke atas setiap fungsi dan kepada sistem secara keseluruhan untuk mengenalpasti ralat-ralat yang wujud.

- **Membuangkan ralat**

Ralat dikeluarkan dengan membuat ralat-ralat yang dikenalpasti semasa pengkompilan (*debugger*).

- **Ujian regresi**

Untuk memastikan bahawa pembetulan/pengeluaran ralat benar-benar menyelesaikannya atau memberi kesan sampingan pada bahagian kod yang lain.

6.2 JENIS-JENIS PENGUJIAN

Terdapat beberapa jenis pengujian yang biasanya dilakukan terhadap sistem bagi mengenalpasti kesemua ralat yang wujud. Pengujian biasanya dilakukan dengan melakukan demonstrasi kesilapan yang ada dalam program. Oleh kerana objektif pengujian adalah untuk mencari kesilapan, maka kesilapan yang ada itu akan diperbaiki dan seterusnya membawa kepada pengujian yang berjaya. Pengenalan kesilapan (*Fault identification*) ialah proses untuk menentukan kesilapan dan jika ada, apakah yang menyebabkan kesilapan tersebut berlaku. Ia diikuti pula dengan pembetulan kesilapan (*Fault correction*) iaitu proses melakukan perubahan terhadap kesilapan yang dikenalpasti.

6.2.1 Pengujian Modul dan Integrasi

Dalam pengujian ini Komponene-komponen modul akan digabungkan untuk mewujudkan satu sistem. Penggabungan ini akan seterusnya memberikan gambaran sebenar apabila berlaku kegagalan sistem. Modul-modul yang digabungkan ini akan berkait diantara satu sama lain, dan suatu kegagalan pada suatu modul akan memberi kesan kepada fungsi modul-modul yang lain. Terdapat 4 jenis kaedah pengujian dalam pengujian modul dan integrasi ini iaitu:

- Integrasi Bawah-Atas (*Bottom-up Integration*)
- Integrasi Atas-Bawah (*Top-down Integration*)
- Integrasi *Big-Bang*
- Integrasi *Sandwich*

Sebelum melakukan pengujian ini pengujian unit (6.2.2) harus dilakukan.

6.2.2 Pengujian Unit

Pengujian unit ialah pengujian ke atas setiap komponen modul aturcara. Modul-modul ini diasingkan daripada modul-modul yang lain supaya ianya dapat diuji secara sendirian.

6.3 JENIS-JENIS KESALAHAN

Setiap sistem akan mempunyai kesalahan yang berbeza-beza. Kesalahan/kesilapan ini boleh dikategori sebagai kesilapan yang ringan dan berat. Apabila wujudnya kesilapan yang berat(susah), sistem akan diuji untuk mengasingkan beberapa banyak kesilapan dengan melakukan pecahan terhadap masalah tersebut kepada kesilapan-kesilapan yang kecil. Apabila kesilapan tersebut telah dipecahkan, pembangun harus mengenalpasti kesilapan tersebut. Kesalahan/kesilapan terdiri daripada 3 jenis:

- Kesalahan Algoritma (*Algorithmic faults*)
- Kesalahan Sintak (*Syntax faults*)
- Kesilapan Dokumentasi (*Documentation faults*)

6.3.1 Kesalahan Algoritma (*Algorithmic faults*)

Kesalahan Algoritma kebiasaannya berlaku apabila algoritma atau logik dalam sesuatu komponen tidak dapat menghasilkan output yang dikehendaki bagi suatu input yang telah dimasukkan. Kesalahan ini berlaku kerana terdapat ralat semasa proses pemprosesan. Kesalahan ini mudah dikenalpasti dengan hanya melihat dan meneliti balik kepada aturcara yang ditulis (*desk checking*). Jenis-jenis kesalahan adalah seperti berikut:

- Terlupa/tersilap untuk mengistiharkan pembolehubah atau gelung berlainan (mengistiharkan pembolehubah integer sebagai string) dan pengistiharan pada kawasan berlainan yang menyebabkan masalah capaian (cth ; public dan private)
- Ujian yang salah untuk syarat pilihan (*testing for the wrong condition*)
- Terlupa untuk menguji terhadap syarat yang tertentu (seperti apabila pembahagian dengan kosong terjadi)

6.3.2 Kesalahan Sintaks (*Syntax faults*)

Kesalahan sintak boleh dikenalpasti semasa berlakunya kesilapan algoritma. Ini akan menyebabkan penulisan suatu bahasa pengaturcaraan tidak membawa maksud yang tepat/sepautnya. Apabila program dijalankan dan berlaku kesalahan sintak, ia dapat dikenalpasti dan akan dimaklumkan jenis kesilapan serta dimana lokasi kesalahan oleh pengkompil.

6.3.3 Kesilapan Dokumentasi (*Documentation faults*)

Proses membuat dokumentasi bagi sesebuah sistem bukanlah suatu kerja yang mudah. Bermacam-macam kesilapan boleh berlaku sebagai contoh kesilapan ejaan, penggunaan istilah yang tidak betul dan susunan ayat yang salah. Kesilapan-kesilapan ini boleh

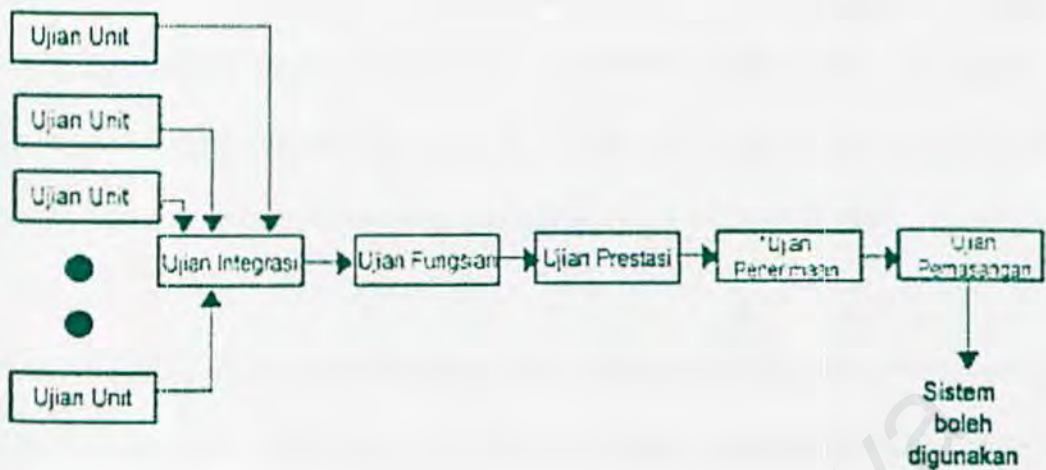
mengakibatkan pemahaman pembaca dokumentasi merosot dan menjadikan dokumen tersebut tidak efektif dalam penyampaian maklumatnya. Dokumentasi sangat penting bagi memastikan sesebuah sistem yang dibangunkan dapat diubahsuai mengikut perubahan semasa dan kesesuaian organisasi.

6.4 PENGUJIAN SISTEM

Di dalam pengujian Sistem Informasi Direktori Bas, terdapat 6 ujian utama yang telah dijalankan. Ujian-ujian tersebut adalah seperti yang dinyatakan dibawah :

- Ujian Unit
- Ujian Integrasi
- Ujian Fungsi
- Ujian Prestasi
- Ujian Penerimaan
- Ujian Pemasangan

Rajah langkah-langkah pengujian yang telah dijalankan pada sistem ditunjukkan di dalam Rajah 6.1. Ujian yang telah dibuat terhadap SIDB merupakan ujian berjenis menaik (*bottom-up testing*) dimana ia pengujian bermula daripada unit-unit terkecil sehinggalah kepada pengujian sistem secara keseluruhan dan menginstall/pemasangan sistem tersebut.



Rajah 6.1: Struktur Pengujian Sistem

6.4.1 Ujian Unit

Ujian unit melibatkan pengujian ke atas unit-unit terkecil yang dikenali sebagai modul. Pengujian dijalankan untuk menguji ketepatan fungsi, logik, syarat sempadan (cth; pengistiharan public & private) dan mengenalpastian ralat. Ia bertujuan untuk memastikan bahawa setiap input yang ditujukan kepada unit-unit tersebut berjalan dengan lancar dan unit-unit memulangkan output yang dikehendaki.

Syarat-syarat sempadan dipastikan dilaksanakan dengan betul berdasarkan keadaan yang telah ditetapkan supaya perpindahan data dapat dilakukan dengan sempurna. Ia juga menguji setiap laluan pengurusan ralat (*error control*) bagi memastikan samada pemprosesan akan diterus dilarikan atau dialihkan ke laluan lain apabila berlaku ralat.

Jadual atau pangkalan data bagi modul ini akan diperiksa untuk memastikan samada data yang dimasukkan melalui antaramuka yang direka berjaya masuk atau tidak ke dalam pangkalan data tersebut. Sekiranya ia berjaya masuk ke dalam pangkalan data maka ia bermakna unit yang dibina bagi modul berkenaan adalah berjaya.

Ujian unit juga dilakukan dengan membaca setiap barisan kod yang telah ditulis semasa membina modul bagi mengenalpasti sebarang kesilapan sintaks, data dan algoritma. Setelah dikenalpasti ia kemudiannya akan dibetulkan oleh pembangun itu sendiri. Ini memerlukan konsentrasi dan tahap penelitian yang tinggi lagi-lagi bila membangunkan sistem yang kompleks. Kemudian, modul ini akan dikompil semula untuk menghapuskan sebarang ralat yang masih tinggal. Kemudian, ia akan dilarikan untuk memastikan ianya berfungsi secara efektif dan secepatimana yang dikehendaki.

6.4.2 Ujian Integrasi

Ujian Integrasi merupakan proses yang memastikan bahawa setiap komponen dalam sistem bekerjasama diantara satu sama secepatimana yang ditetapkan spesifikasi. Ia dijalankan kepada satu atau lebih modul yang telah digabungkan. Kemudian data-data yang sesuai akan dimasukkan ke dalam modul-modul dan ujian. Kemudian pangkalan data modul-modul ini akan diperiksa untuk menjamin penghantaran data dari satu modul ke modul yang lain tidak hilangserta memastikan juga penyepaduan modul tidak memberi kesan negatif terhadap prestasi modul sendiri mahupun yang lain.

Dalam menjalani pengujian integrasi bagi sistem ini, ujian integrasi bawah atas (*bottom-up*) digunakan sebagai landasan. Dalam strategi ini, komponen yang wujud di tahap paling rendah di dalam hierarki sistem akan diuji terlebih dahulu. Ia akan diuji secara individu tanpa memberi kesan kepada modul lain (tidak terikat). Kemudian, komponen yang akan diuji ialah komponen yang memanggil sub-komponen dibawahnya iaitu yang telah diuji tadi. Proses ini akan diteruskan sehingga ke hierarki teratas (komponen teratas dalam hierarki) telah diuji.

6.4.3 Ujian Fungsi

Setelah maklumat yang dihantar di antara komponen bertepatan atau mengikut rekabentuk spesifikasi, sistem akan diuji bagi memastikan ianya memenuhi Keperluan fungsi yang diinginkan. Ujian fungsi menilai sistem bagi menentukan fungsi-fungsi yang telah diuraikan oleh spesifikasi keperluan betul-betul dipersembahkan dalam sistem yang telah dibangunkan pembangun.

Ujian yang dijalankan juga melibatkan ujian baikpulih (*Recovery testing*) yang bertujuan untuk memastikan bahawa jika kegagalan sistem berlaku ia dapat dipulihkan semula samada secara automatik (error control) oleh sistem atau berdasarkan masukan input daripada pengguna (reset/input nilai kawalan).

6.4.4 Ujian Prestasi

Ujian dijalankan pada masa larian (*run time*) bagi memastikan prestasi sistem secara keseluruhannya dari segi antaramuka dan fungsian. Ia juga digunakan untuk menguji aspek-aspek prestasi program terhadap komputer pengguna seperti masa tindakbalas pemprosesan, kecekapan sistem dan ingatan yang digunakan. Apabila ujian telah dijalankan dengan betul, sistem telah berjaya berfungsi di dalam persekitaran sebenar. Oleh itu, sistem yang dibina adalah sah dan boleh dipercayai.

6.4.5 Ujian Penerimaan

Setelah ujian prestasi dijalankan dengan sempurna, langkah seterusnya melibatkan pengguna. Pengguna akan memastikan bahawa sistem yang telah dibangunkan menepati kehendak dan jangkaan pengguna. Ia mestilah berfungsi mengikut kehendak pengguna.

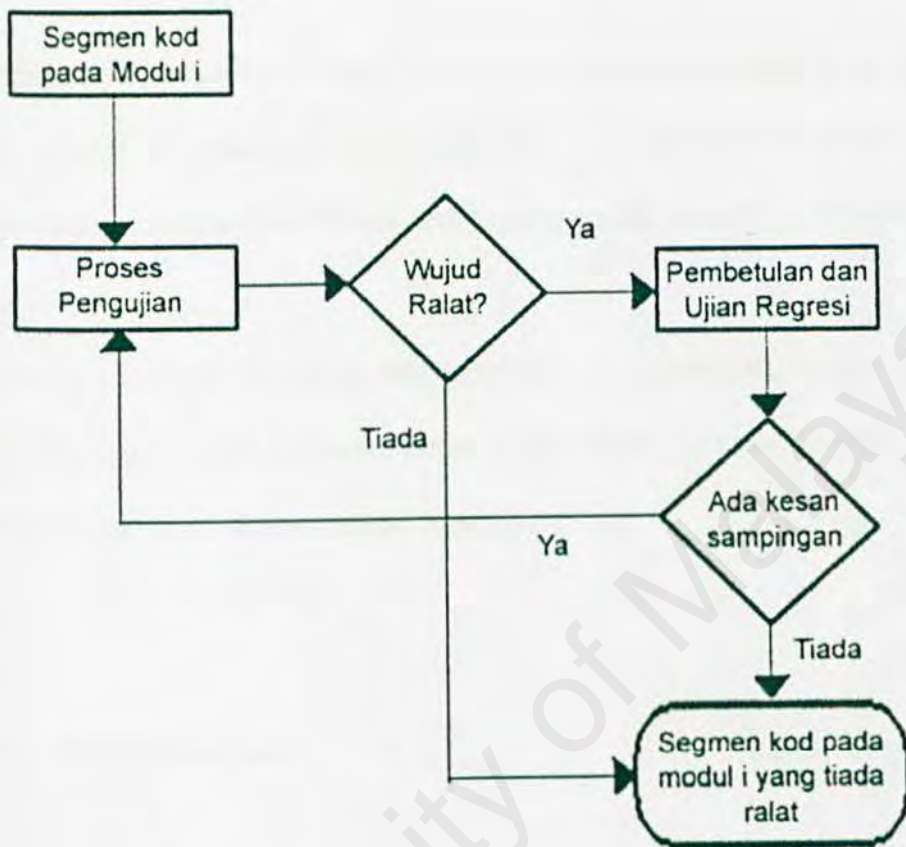
6.4.6 Ujian Pemasangan

Setelah sistem melepasi ujian penerimaan, sistem akan diuji dengan memasangnya/menginstall di persekitaran sebenar (*working-environment* di mana ia akan digunakan. Secara kasarnya, langkah utama yang terlibat di dalam ujian ini ialah

pembangun akan cuba memasang sistem yang telah dibina di persekitaran sebenar untuk memastikan sistem boleh berjalan dengan lancar. Sekiranya ia gagal untuk berfungsi seperti yang diinginkan, maka pembangun perlu melakukan beberapa pengubahsuaian pada sistem atau persekitaran sistem agar sistem boleh berfungsi sebaik yang mungkin dalam persekitaran tersebut.

6.5 TEKNIK PENGUJIAN SISTEM

Teknik yang digunakan adalah mengikut peringkat ujian yang telah ditetapkan. Pada peringkat pengujian unit, teknik kotak putih (*white box*) telah digunakan untuk menentukan sebarang ralat seperti yang ditunjukkan dalam Rajah 6.2.



Rajah 6.2: Ujian Unit

6.6 MEREKA FAIL EXECUTABLE DAN SET-UP

Setelah program telah diuji dengan teliti, pembangun akan menukar program yang telah dibina kepada fail *executable* iaitu fail berjenis .exe. Perisian Visual Basic mempunyai fungsi dalam menghasilkan fail *executable* yang mudah untuk diimplementasikan.

Fail *set-up* juga perlu dihasilkan bagi membolehkan sistem yang dibina dapat dilarikan dalam komputer yang berlainan tanpa memerlukan komputer tersebut mempunyai perisian Visual Basic dan tambahan "component".

6.7 KESIMPULAN

Dengan ujian-ujian yang telah dilaksanakan, ralat-ralat yang wujud dalam sistem ini dapat dikenalpasti dan kemudian diselesaikan dengan baik. Walaubagaimanapun, ini tidak bermakna sistem ini sudah bersih dan tiada lagi sebarang ralat padanya. Tetapi sekurang-kurangnya ini dapat mengurangkan risiko pada sistem. Selain itu, kesemua fungsian dan kelakuan sistem dapat dipastikan telah memenuhi keperluan sistem.

BAB 7 PENYELENGGARAAN DAN PENILAIAN SISTEM

7.1 PENYELENGGARAAN SISTEM

Dalam bab ini kita akan membincang mengenai keperluan penyelenggaraan sistem.

Proses ini dilakukan untuk memberikan panduan dan pemahaman kepada pengguna untuk menjalankan penyelenggaraan terhadap sistem agar ianya tidak menjejaskan pengoperasian sistem secara keseluruhan. Bab ini juga akan membincangkan bagaimana penyelenggaraan sistem dapat dilakukan apabila keperluan fungsian semakin berubah. Kaedah pengembalian bencana untuk sistem ini juga akan dibincangkan secara am.

7.1.1 Keperluan Penyelenggaraan

Secara amnya, sesebuah sistem harus diselenggarakan dari semasa ke semasa untuk memastikan sistem beroperasi pada tahap yang optimum. Penyelenggaraan perlu dilakukan ke atas SIDB ini kerana beberapa sebab iaitu:

- **Kandungan Data**

Data-data dalam Senarai XML, perlulah dijaga agar tiada sebarang kerosakan yang boleh mengurangkan keberkesanan data kepada sistem. Kerosakan ini boleh terjadi akibat penghapusan data secara tidak sengaja, penghapusan syntax kawalan, dan data tak lengkap. Data-data dalam XML juga harus dipelihara daripada “*redundancy*”, iaitu pertindihan data yang lama dengan data yang baru dimasukkan.

7.1.2 Metodologi Penyelenggaraan Sistem

Sistem ini dapat diselenggarakan melalui pelbagai kaedah dan antaranya ialah :

- **Penambahan Saiz Storan**

Penambahan saiz storan bagi data-data berbentuk XML dapat dilakukan dengan mudah kerana SIDB adalah “*independent*” daripada fail XML yang digunakan untuk memperoleh data laluan bas.

7.1.3 Pelan Pengembalian Bencana

Pelan pengembalian bencana bertujuan untuk menyediakan sandaran kepada pengoperasian sistem sekiranya berlaku bencana. Kandungan asas sistem dan fail XML (mengandungi data laluan bas) boleh disalin ke dalam media storan lain seperti cakera padat, disket dan sebagainya. Sekiranya berlaku sebarang bencana, sistem masih boleh digunakan dan tidak perlu dibangunkan semula. Jadi, sebagai langkah berjaga-jaga back-up telah dilaksanakan. Kandungan fail XML juga perlu disalin ke media storan dari semasa ke semasa bagi memastikan ianya sentiasa diperbaharui. Data XML tidaklah memerlukan storan yang besar, hanya memadai dengan satu disket sahaja bagi satu kawasan laluan yang tertentu.

7.2 PENILAIAN SISTEM

7.2.1 Pendahuluan

Salah satu cara terbaik untuk membangunkan sesebuah sistem ialah dengan melibatkan fasa penilaian sistem di dalam kitar pembangunan sistem itu. Fasa ini adalah bertujuan agar pembangun dapat menganalisa sejauh manakah kejayaan sistem yang dibangunkan dapat mencapai objektifnya. Biasanya pembangun menerima maklumbalas pengguna dalam menilai sesebuah sistem.

7.2.2 Pencapaian Objektif

Sistem ini secara keseluruhannya hampir berjaya mencapai objektif utama yang telah digariskan semasa fasa analisis dijalankan. Namun terdapat fungsian serta objektif yang tidak dapat direalisasikan oleh pembangun. Sebab utama fungsian serta objektif ini tidak dapat dilaksanakan ialah:-

- Pertama, tiada sumber maklumat yang efisien dan tepat mengenai laluan bas yang sedia ada, ini telah dikenalpasti sebagai punca utama kegagalan pembangunan sebahagian fungsi dan pencapaian objektif secara menyeluruh. Dengan kekurangan ini, sistem ini tidak dapat mencapai 100% objektif yang telah disasarkan. Namun, tanpa sumber tepat mengenai laluan bas ini, sistem

dapat juga berfungsi baik dengan menggunakan laluan rekaan yang seakan laluan benar. Laluan rekaan ini tidaklah direka semata-mata tetapi direka dengan maklumat yang minimum.

- Kedua, sistem tidak dapat dibangunkan dalam bentuk “on-line” kerana pembangun tidak mempunyai masa yang secukupnya bagi mengimplementasikan bahagian ini. Ini kerana, fungsi lain yang lebih penting dalam objektif seperti algoritma laluan terpantas memakan masa yang lama untuk direalisasikan. Pengkodan dan cara algoritma berfungsi adalah agak kompleks dan memakan masa yang lama untuk difahami. Sumber rujukan yang diperlukan untuk algoritma ini juga agak sukar untuk diperolehi. Hanya beberapa sumber sahaja yang menepati kehendak pembangun dan ianya adalah agak kompleks. Jadi dengan ini, tajuk sistem yang dibangunkan juga telah berubah menjadi “*SISTEM INFORMASI DIREKTORI BAS*”. Nama sebelum ini dianggap telah diganti dengan nama baru ini bersesuaian dengan fungsinya. SIDB ialah nama singkatan bagi nama baru ini.

SISTEM INFORMASI DIREKTORI BAS

Rajah 7.1: Nama Baru Sistem

7.2.3 Masalah dan Penyelesaian

Terdapat pelbagai masalah yang dihadapi oleh pembangun dalam membangunkan sistem ini. Sesetengah masalah dapat diselesaikan dengan mudah manakala terdapat juga masalah sukar diselesaikan. Namun pembangun telah berjaya menyelesaikan masalah-masalah yang mudah tersebut dengan bantuan pelbagai pihak secara langsung dan tidak langsung. Terdapat beberapa kategori masalah yang telah dikenalpasti seperti yang diuraikan di bawah:

7.2.3.1 Masalah Maklumat Laluan Bas Terkini

Di dalam proses pembangunan sistem ini, maklumat laluan bas yang terkini tidak dapat diperolehi kerana sebab-sebab yang akan diterangkan dibawah. Maklumat bas ini adalah peta yang memaparkan laluan bas di kawasan Petaling Jaya. Sedia maklum bahawa kawasan laluan yang dicadangkan dahulu iaitu keseluruhan kawasan Kuala Lumpur dan Petaling Jaya tidak dapat digunakan oleh kerana kekurangan maklumat. Bagi syarikat bas yang mengeluarkan peta ini, maklumat ini dianggap terhad. Oleh itu, ia sememangnya sukar untuk diperolehi, dan jika ingin memperolehinya perlu mendapat kebenaran dari pihak yang berkenaan. Tetapi, setelah melawat laman web syarikat bas tersebut, mereka ada menyediakan link kepada peta laluan bas. Peta laluan yang disediakan adalah terlalu am dan mewakili keseluruhan kawasan perkhidmatan. Oleh itu, peta laluan terpaksa direka dengan menggunakan peta jalan seta maklumat yang

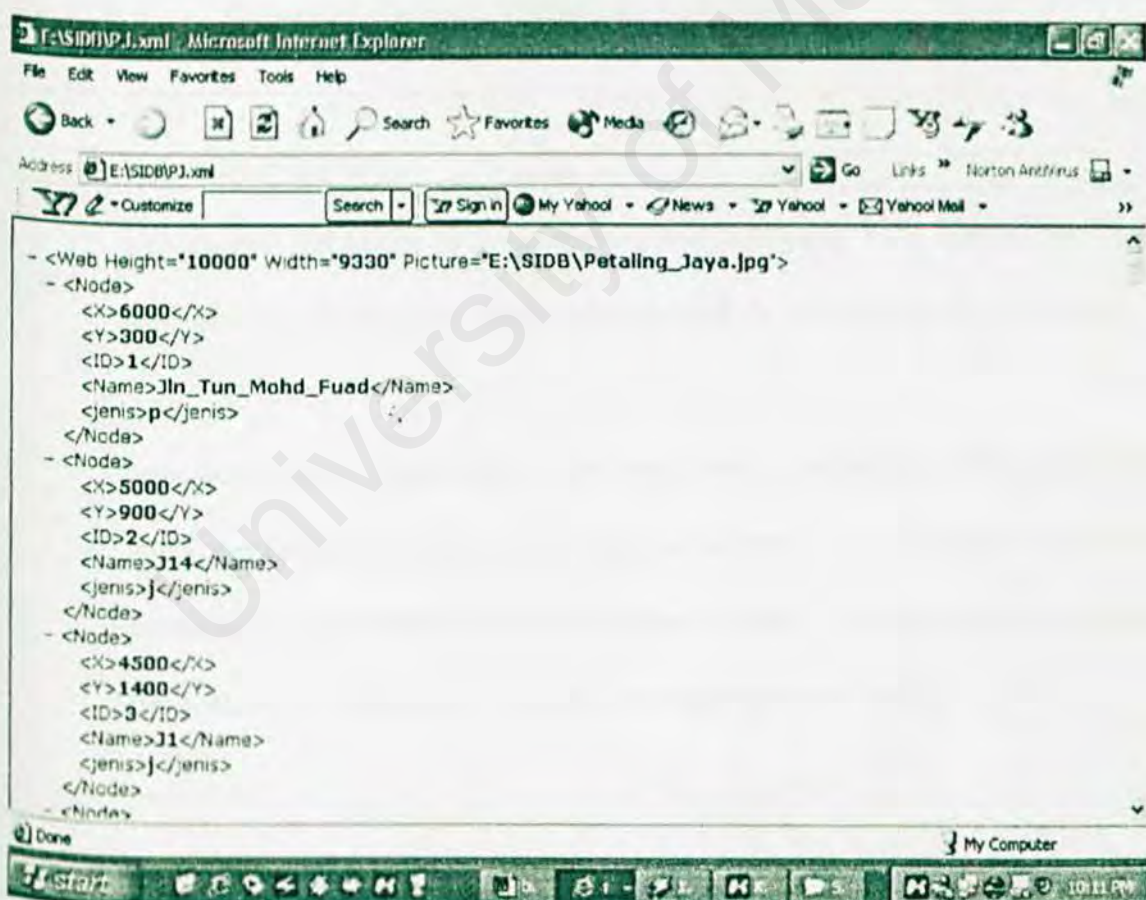
diperolehi daripada jadual perjalanan bas. Jadual ini diperolehi daripada laman web lama syarikat bas tersebut. Jadual ini tidaklah beberapa terkini (*updated*). Peta jalan yang diperolehi juga tidak beberapa lengkap dari segi kuantiti serta kualiti jalan yang dipaparkan (kebanyakannya memaparkan jalan-jalan utama; mengabaikan jalan-jalan lain yang dilalui bas). Adakalanya nama jalan tidak ditunjukkan dan tidak dikemaskinikan. Ini menyukarkan proses mentafsir maklumat laluan bas yang terkini. Oleh itu, laluan bas rekaan yang digunakan ini tidaklah menunjukkan keadaan sebenar laluan bas di kawasan Petaling Jaya dan ini menyebabkan paparan peta yang ditunjukkan dalam sistem tidak dapat dipopulasikan sepenuhnya. Ini seterusnya mengurangkan keberkesanan sistem dalam persekitaran sebenar.

7.2.3.2 Masalah Pangkalan Data

Perubahan dari segi perisian pangkalan data telah dilakukan, dimana pembangun telah mengabaikan penggunaan Microsoft Access 2002 untuk mencipta pangkalan data tetapi telah menggunakan bahasa pengaturcaraan XML. Sebab mengapa pangkalan data Access 2002 tidak digunakan ialah :-

- Microsoft Access 2002 tidak mempunyai integrasi yang begitu baik dengan Visual Basic 6 dan ini melambatkan proses pembangunan. Microsoft Access 97 Menunjukkan tahap kesesuaian (compatibility) yang tinggi dengan VB6 tetapi malangnya ia tidak dapat digunakan kerana tidak bersesuaian dengan sistem operasi (OS) komputer pembangun. Jika dilihat secara meluas, pada masa kini hampir kesemua sistem komputer yang baru menggunakan sistem operasi yang terbaru, dan ini menjadi kebimbangan kepada pembangun untuk menggunakan Access 97. Masalah tentu akan berlaku lama-kelamaan dengan pangkalan data Access 97 jika ianya perlu diubahsuai atau dikemaskinikan. Dengan menggunakan XML, isu "*compatibility*" tidak timbul dan ianya adalah lebih senang diakses serta lebih mudah diintegrasikan.
- Access 2002 kurang sesuai untuk mempersembahkan data melalui web atau online. XML sebaliknya adalah mudah.
- Dokumen XML adalah mudah difahami dan jelas. Jika pengguna tidak mempunyai pelayar XML (masih belum lagi wujud; peringkat awal), seseorang dapat melihat isi kandungan dokumen XML tersebut dengan hanya menggunakan "text editor" dan seterusnya memahami maksud yang ingin dibawa dengan mudah. Ini menjadikan penggunaan XML lebih mesra pengguna dan senang diubahsuai.
- Dokumen XML boleh disediakan dengan pantas. Seterusnya memudahkan sebarang perubahan dan penciptaan data baru.

- Rekaan dokumen XML adalah dilakukan dengan formal dan konsisten. XML kurang “standards” dimana ia tidak mempunyai banyak “pre-defined tags” seperti HTML. Ini menjadikan XML “highly flexible”.
- Dokumen XML adalah mudah untuk direka. Walaupun masih tidak wujud lagi “editors” untuk mereka dan mengubahsuai kandungan XML, ia masih mudah untuk berbuat sedemikian dengan hanya menggunakan “text editor”.



Rajah 7.2: Paparan XML pada pelayar *Internet Explorer 6.0*. XML memerlukan pelayar

XML untuk melihat kandungannya dengan betul.

```

<Web Height="10000" Width="9330" Picture="E:\SIDB\Petaling_Jaya.jpg">
<Node>
  <X>6000</X>
  <Y>300</Y>
  <ID>1</ID>
  <Name>Jln_Tun_Mohd_Fuad</Name>
  <jenis>p</jenis>
</Node>

```

Rajah 7.3: Contoh dokumen XML.

7.2.3.3 Masalah Dalam Rekabentuk Sistem

- Kekurangan kemahiran

Kekurangan kemahiran di dalam bahasa pengaturcaraan Visual Basic dan XML menyebabkan pembangun menghadapi masalah dalam memulakan projek ini.

Namun, akhirnya pembangun dapat mengatasi masalah ini sedikit sebanyak untuk merealisasikan sistem ini. Tetapi, masa yang diambil adalah lama kerana pembangun terpaksa bermula daripada asas. Pembangun juga banyak mendapat bantuan daripada rakan-rakan sebaya dan juga rakan dari internet.

❑ Kekurangan bahan rujukan

Pada mulanya, pembangun telah menghadapi masalah kekurangan sumber rujukan untuk pembangunan sistem ini. Sumber rujukan yang terdapat di pasaran adalah sangat mahal dan buku rujukan yang terdapat di perpustakaan adalah sangat terhad dan tidak terkini.

Masalah ini dapat diatasi oleh pembangun dengan mengambil beberapa alternatif seperti meminjam buku-buku berkaitan daripada rakan-rakan selain terpaksa membeli buku-buku yang sesuai untuk pembangunan sistem. Selain itu, internet juga memberikan maklumat yang berguna dan banyak mengenai sistem yang ingin dibangunkan. Tidak ketinggalan juga, forum-forum perbincangan mengenai Visual Basic di dalam internet yang turut disertai oleh pembangun.

❑ Masalah menghubungkan perisian

Pada permulaan proses pembinaan sistem, pembangun menghadapi masalah dalam menghubungkan antaramuka pengguna Visual Basic dengan XML.

Masalah Ini dapat diatasi dengan membuat rujukan kepada buku-buku berkenaan dan juga dengan melayari internet serta mencari contoh penyelesaian bagi masalah yang serupa.

7.2.4 Maklumbalas Pengguna

Sebenarnya, Sistem Informasi Direktori Bas yang dibina masih belum melepasi tahap penilaian oleh pengguna akhir (*end user*) sepenuhnya. Ini adalah kerana kesuntukkan masa yang tidak dapat dielakkan. Hanya segelintir pengguna telah berpeluang menggunakan sistem ini. Pengguna yang berpeluang ini terdiri daripada rakan-rakan dan ahli keluarga. Penilaian oleh golongan pengguna yang kerap menggunakan perkhidmatan bas tidak dapat dilakukan dengan meluas.

Secara keseluruhannya, penguji sistem kebanyakannya tidak pernah menggunakan program yang seakan-akan Sistem Informasi Direktori Bas. Jadi, pengujian sistem ini menjadi suatu pengalaman baru bagi mereka. Penguji yang ada telah dibahagikan kepada dua kategori:

- Penguji biasa (tiada latarbelakang sains komputer)
- Penguji berlatarbelakangkan sains komputer

Penguji biasa yang telah menguji sistem secara amnya akan mengfokus kepada bahagian antaramuka iaitu persembahan dan fungsi yang sistem sediakan. Mereka memberi input/ulasan yang berguna dalam bagaimana sistem dapat dibangunkan supaya ia lebih menarik dari segi persembahan. Ada diantara mereka terutamanya pengguna bas yang tetap telah sedikit-sebanyak mencadangkan fungsi-fungsi yang boleh ditambah

supaya sistem lebih menarik. Tetapi, sepertimana yang dikehendaki proses pengujian, mereka telah menguji sistem supaya berfungsi sepertimana yang dinyatakan dalam spesifikasi dan sesuai digunakan dalam “*working-environment*”.

Sebagai langkah tambahan, iaitu bagi memastikan Sistem Informasi Direktori Bas yang dibangunkan ini berkesan, pengujian dengan pengguna telah juga dilakukan dengan pengguna yang mempunyai latar belakang pendidikan di bidang sains komputer. Pengujian ini adalah berbeza dimana pengguna yang mempunyai latar belakang pendidikan sains komputer dikehendaki menguji dan menilai aspek-aspek persembahan data, implimentasi algoritma, susunan borang (*form*), penggunaan ciri-ciri tambahan seperti penggunaan “*toolbar*”. Antara maklumat yang diberi oleh penguji sistem adalah hampir sam dengan penguji biasa tetapi dengan beberapa kelebihan seperti :

- Kekurangan Sistem
- Kelebihan Sistem
- Komen mengenai pemilihan antaramuka pengguna
- Komen mengenai susunan borang (*form*)
- Ulasan dan cadangan terhadap sistem

Penguji berpuas hati dan merasa selesa dengan pengantaramuka pengguna yang digunakan. Bilangan borang yang minima telah digunakan untuk memudahkan pemahaman pengguna. Borang yang betul-betul berkesan dalam penyampaian maklumat ditekankan susunannya supaya memberi kesan yang maksima. Susunan

borang (*form*) adalah tersusun dan mudah difahami. Secara keseluruhannya, semua penguji mendapati bahawa sistem serta perjalanannya memuaskan.

7.2.5 Kekuatan Sistem

Berbanding dengan sistem manual, Sistem Informasi Direktori Bas ini mempunyai kelebihan tersendiri :

7.2.5.1 Lebih Efektif

Sistem manual yang digunakan oleh hampir kesemua pengguna bas memerlukan pengetahuan yang mendalam mengenai laluan bas dan boleh mengambil masa yang lama untuk memperolehi keputusan. Adakalanya keputusan yang diperolehi tidaklah begitu tepat. Kelemahan sistem manual ini telah banyak dibincangkan dalam Bab 2. Dengan menggunakan SIDB ini pengguna dapat berinteraksi dengan sistem dan memperolehi laluan bas yang terpantas dengan cepat dan tepat. Kesimpulannya, SIDB telah banyak menyelesaikan masalah yang dihadapi sistem manual.

7.2.5.2 Ciri Keselamatan

Sistem ini hanya membenarkan pengguna berdaftar yang mempunyai nama login dan katalaluan yang sah sahaja untuk melarikan sistem ini.

7.2.5.3 Mesra pengguna

Sistem ini boleh digunakan oleh bukan sahaja oleh pengguna yang mahir dalam penggunaan komputer sahaja, malah pengguna yang langsung tidak mempunyai latar belakang berkaitan komputer juga mampu menggunakannya. Ini adalah kerana penggunaan butang-butang "*command*" bagi fungsi yang bersesuaian memudahkan pengguna menggunakan sistem dan ianya mudah difahami. Sistem juga mempunyai tahap interaksi yang tinggi dimana sistem memaparkan suatu peta interaktif yang mana pengguna boleh mencari lalaun terpantas dengan hanya point & click pada peta. Penggunaan bahasa Melayu dalam sistem ini amat memudahkan semua pihak terutamanya pengguna untuk menggunakan sistem ini. Pengguna bahasa melayu juga akan memberi kelebihan dan identiti unik dimana semua bangsa di Malaysia boleh menggunakannya. Selain itu, sistem ini juga menyediakan antaramuka yang kelihatan professional dan menarik.

7.2.5.4 Lebih Fleksibel

SIDB juga menawarkan ciri kefleksibelan yang tinggi dengan kata lain menyediakan ciri-ciri pengemaskinian yang tinggi. Pengguna boleh membawa atau mencipta data-data laluan bas mereka sendiri dengan mudah. Data-data yang dicipta ini dapat digunakan oleh sistem dengan mudah. Pengemaskini adalah terhad dan tidak semua pengguna boleh melakukannya; ini bertujuan untuk memelihara integrasi dan keberkesanan sistem. Hanya pengguna yang mengetahui pengetahuan XML, pengetahuan mendalam mengenai laluan bas serta sumber data yang mencekupi boleh mengemaskinkan sistem. Pengemaskinian boleh dilakukan kepada hampir semua aspek data laluan bas serta peta yang digunakan sebagai rujukan laluan bas.

7.2.5.5 Trend Semasa

Pada masa kini, dalam era Teknologi Maklumat adalah amat menguntungkan dan efektif sekiranya sistem pengkomputeran seperti SIDB diaplikasikan di kawasan-kawasan tertentu terutamanya kawasan awam. Ini adalah kerana sistem seperti ini mampu menerima dan memproses input dengan cepat dan efektif dengan hanya input pengguna yang minima. SIDB juga menyediakan antaramuka menarik dan ini menjadi tarikan kepada pengguna untuk menggunakannya.

7.2.6 Kekangan Sistem

Walaupun sistem yang dibina mempunyai beberapa kekuatan, namun ia juga tidak lari daripada kekangan tertentu. Antara kekangan-kekangan sistem yang dapat dikenalpasti oleh pembangun ialah :

7.2.6.1 Sistem Pengemaskinian Terhad

Oleh kerana tempoh masa untuk pembinaan sistem ini agak singkat, maka pembangun tidak sempat untuk mencipta suatu sistem pengemaskinian On-line bagi SIDB (sepertimana yang dibincangkan didalam sub-bab 7.2.2: Pencapaian Objektif). Sistem pengemaskinian ini secara amnya akan memuat-turunkan fail XML yang akan mengemaskinkan data-data laluan yang terkini.

7.2.6.2 Pengujian Penerimaan

Sebenarnya SIDB yang dibina masih belum melepasi pengujian penerimaan yang sebenar. Ia adalah kerana kesuntukkan masa yang tidak dapat dielakkan. Ini menyebabkan pembangun tidak sempat mendapat kebenaran untuk mengujinya di tempat-tempat awam seperti perpustakaan dan kepada golongan pengguna bas yang lebih meluas.

7.2.7 Pembaharuan Masa Hadapan

Sistem SIDB yang dibina ini masih boleh diperbaharui supaya ianya lebih efisien dan efektif. Banyak fungsi serta idea tambahan yang dapat diwujudkan pada sistem tetapi masa tidak mengizinkan bagi merealisasikan fungsi-fungsi tersebut. Misalnya, perancangan perjalanan pintar dimana pengguna hanya perlu memasukkan lokasi dan destinasi dan sistem akan memaparkan kesemua maklumat yang berkaitan; sebagai contoh bas apa yang perlu dinaiki, pertukaran bas yang perlu, waktu paling sesuai untuk perjalanan tersebut dan tambang yang akan dikenakan, semua ini menjurus ke perjalanan bas yang paling pantas. Ini secara tak langsung, memudahkan perancangan perjalanan seseorang serta banyak mengurangkan masa perancangan. Sistem juga dilihat dapat mengemaskinikan dirinya sendiri sebaik sahaja maklumat laluan bas yang terkini wujud. Ia akan dilakukan secara automatik melalui prinsip atas talian. Ini dapat mengurangkan masa pengemaskinian yang perlu dilakukan pengguna terutamanya bagi pengguna yang tidak tahu bahasa XML. Pengemaskinian akan dilakukan kepada data dalam XML dan peta (imej). XML dan peta yang berbaza dapat megurangkan keberkesanan dan keefisienan sistem, pengguna perlu mengemaskinikan data XML dan peta berasingan, jadi pada masa hadapan, XML dan peta akan digabungkan sekali dan ini akan menghasilkan suatu peta interaktif yang mana pengguna boleh memilih lokasi dan destinasi diatas peta interaktif tersebut.

Selain daripada itu pembangun juga akan membangunkan sistem ini secara "*on-line*" sebagaimana yang telah nyatakan dalam spesifikasi suatu masa dahulu. Apabila sistem secara "*on-line*" diimplementasikan, sistem dapat diakses melalui web. Dengan kebolehan akses dari web ini, sistem dapat dikemaskini dari segi data, serta diubahsuai melalui "*patch*" atau "*update*". Bahasa pengaturcaraan yang lebih efisien dan sesuai akan digunakan bagi pembangunan web kelak (cth. bahasa pengaturcaraan yang sesuai ialah VB script dan XML).

7.3 Rumusan

Daripada bab ini, segala masalah dan penyelesaian yang telah diambil oleh pembangun telah dibincangkan dengan teliti. Selain itu penilaian-penilaian lain juga telah dititikberatkan. Walaupun wujud beberapa kekangan namun dapat ditegaskan di sini bahawa SIDB telah mencapai objektif utama.

KESIMPULAN

Secara ringkasnya tujuan pembangunan sistem ini adalah untuk memberi satu kaedah penyebaran maklumat yang lebih berkesan berasaskan teknologi komputer. Ianya bertujuan untuk menggantikan sistem manual seperti papan tanda, buku panduan dan peta laluan bas yang digunakan sekarang. Penggunaan sistem yang sedia ada tidak membawa kesan yang meluas kepada pengguna berbanding dengan sistem yang akan dibangunkan, secara atas-talian, dimana ia boleh dicapai dimana-mana sahaja dan sedia ada. Sistem juga dapat membuat rancangan perjalanan berdasarkan input pengguna. Mereka tidak perlu lagi mengkaji peta atau bertanya. Walaupun sistem yang akan dibangunkan tidak akan menggantikan sepenuhnya penggunaan sistem sedia ada, ia diharap akan dapat digunakan secara selari dengan sistem manual. Sistem manual masih mempunyai kelebihan yang tersendiri.

Segala pengalaman yang diperolehi semasa pembangunan sistem ini akan dapat menambahkan lagi pengetahuan yang sedia ada dalam pelbagai aspek pembangunan sistem. Pembangunan sistem dengan menggunakan metodologi (Kitar Hayat) memberi garis panduan yang sistematik dan mudah. Perancangan yang teliti dapat dilaksanakan sebelum produk akhir lengkap. Penilaian yang intensif telah dijalankan pada sistem untuk memastikan bahawa sistem berfungsi mengikut objektif yang ditetapkan. Sistem secara keseluruhan telah mencapai objektif utamanya. Akhir sekali, diharapkan sistem yang akan dibangunkan nanti dapat memenuhi keperluan pengguna dan dapat memberi manfaat sepenuhnya kepada semua pengguna.

RUJUKAN

Adobe Photoshop Home Page

http://www.adobe.com/products/photoshop/main.html?awe_5710

Microsoft Visual InterDev Home Page

<http://msdn.microsoft.com/vinterdev/default.asp>

Microsoft Visual Visual Basic Home Page

<http://msdn.microsoft.com/vstudio/>

Microsoft Access Home Page

<http://www.microsoft.com/office/access/>

A* Algorithm

http://ugrad-www.cs.colorado.edu/~cs3202/papers/Thien_Than.html

Dijkstra Algorithm

<http://www.csulb.edu/~fnewberg/PCTMSummary/FinalPDFs/jaeson.PDF>

Heuristic Search

<http://www.cs.nott.ac.uk/~nza/G5BADs/slides17.pdf>

Dijkstra's Algorithm

<http://www.deakin.edu.au/~agoodman/graph/dijkstra1.htm>

Pathfinding: A Comparison of Algorithms

http://cpcug.org/user/scifair/Preygel/Preygel.html#_Toc445443577

An optimal pathfinder for vehicles in real-world digital terrain maps

<http://www.student.nada.kth.se/~f93-maj/pathfinder/4.html>

Fastest path problems in dynamic transportation networks

<http://www.husdal.com/mscgis/research.htm>

Intrakota

<http://www.suarasegambut.com.my/intrakota.htm>

Transports Information System of Madrid

<http://www.ctm-madrid.es/sit/SitPpalE.htm>

A Web-based Bus Information System

<http://www.esri.com/library/userconf/proc99/proceed/papers/pap494/p494.htm>

XMLPitstop.com

<http://www.xmlpitstop.com/>

O'Reilly XML.com

<http://www.xml.com/>

Microsoft Developer Network

<http://msdn.microsoft.com/>

Sean, James, A. 1989. *"Analysis & Design of Information system"*, Georgia State University (USA): McGraw-Hill Book Co.

Rajaraman, V. 1991. *"Analysis & Design of Information system"*, New Delhi: Prentice – Hall of India.

Pfleeger, Shari, L. 2001. *"Software Engineering – Theory and Practice"*, Second Edition, Prentice Hall International Inc.

Deitel, H.M. Deitel P.J. Neito, T.R. 1999. *"Visual Basic 6 How To Program"*, Prentice Hall International Inc.